離散最適化基礎論

ジョブ・スケジューリングのアルゴリズム

第 7 回

先行制約：基礎

岡本 吉央　(電気通信大学)

okamotoy@uec.ac.jp

2024 年 11 月 26 日

最終更新：2024 年 11 月 27 日　08:23

スケジュール (前半)

1. スケジューリング問題の分類 (10/1)
 * 休み (出張) (10/8)
 * 休み (体育祭) (10/15)
2. 整列による解法 (10/22)
3. 動的計画法 (10/29)
4. NP 困難性と計算量の分類 (11/5)
5. 計算複雑性による問題の分類 (11/12)
6. リスト・スケジューリング (11/19)

スケジュール (後半)

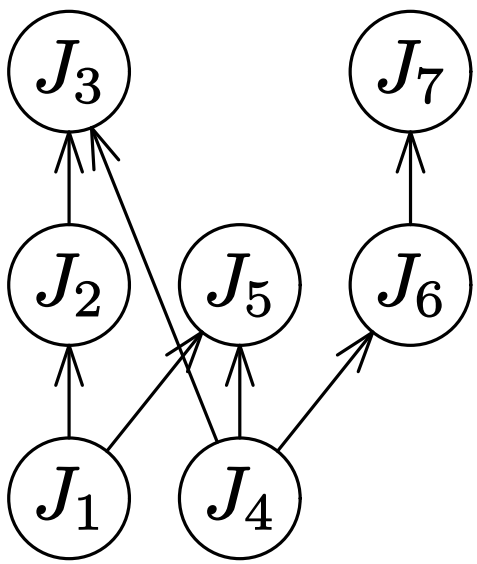
7.	**先行制約：基礎**	(11/26)
＊	休み (秋ターム試験)	(12/3)
8.	先行制約：多機械	(12/10)
9.	先行制約：他の半順序	(12/17)
10.	ショップ・スケジューリング：基礎	(12/24)
＊	休み (冬季休業)	(12/31)
11.	ショップ・スケジューリング：機械数が定数	(1/7)
12.	ショップ・スケジューリング：機械数が可変	(1/14)
13.	近似可能性と近似不可能性	(1/21)
14.	多項式時間近似スキーム	(1/28)
＊	なし	(2/4)

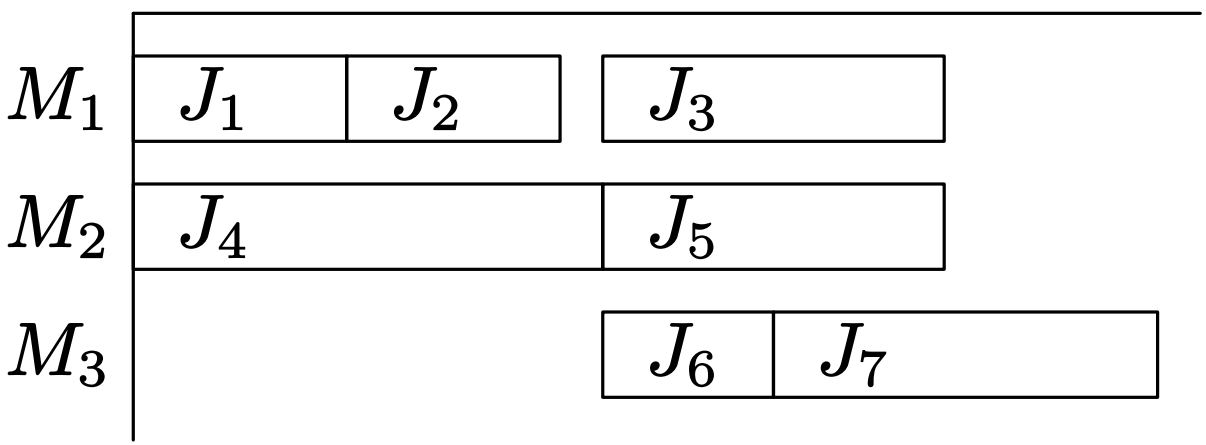
復習：先行制約

ジョブの特性の１つ：先行制約 (precedence constraint)

ジョブの集合 $\mathcal{J}$ 上の半順序 $\rightarrow$ を使って
J_j の処理が完了しないと $J_{j'}$ の処理を開始できない
ことを $J_j \rightarrow J_{j'}$ で表す

先行制約があるとき，β に「prec」と書く　　(precedence)





今日の内容

目標

ジョブ・スケジューリングにおける
先行制約の扱い方 を学び, それを正しく使えるようになる

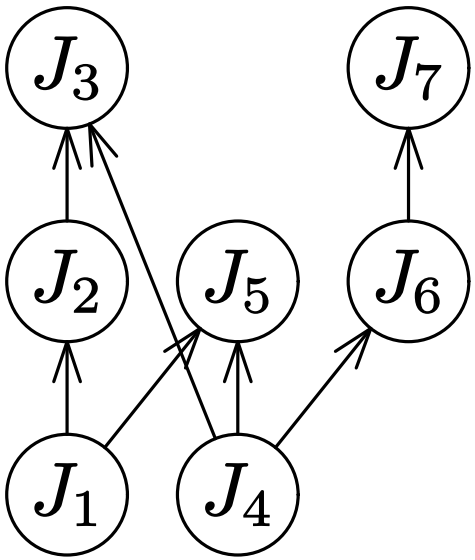
- 重要概念：トポロジカル順序，クリティカル・パス
- 主に対象とする問題
  - $1 \mid \mathrm{prec} \mid L_{\max}$
  - $1 \mid \mathrm{prec} \mid \sum U_j$
  - $P\infty \mid \mathrm{prec} \mid C_{\max}$

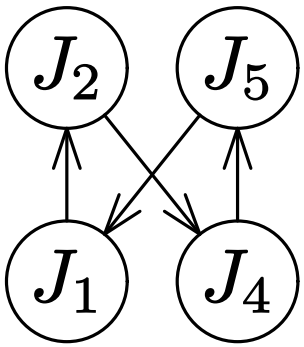
本日の内容

1. **先行制約の基礎概念**
2. 一機械スケジューリングと先行制約
3. $P\infty \mid \mathrm{prec} \mid C_{\max}$ とクリティカル・パス法

- A. B. Kahn, Topological sorting of large networks. *Communications of the ACM* 5 (1962) pp. 558–562.

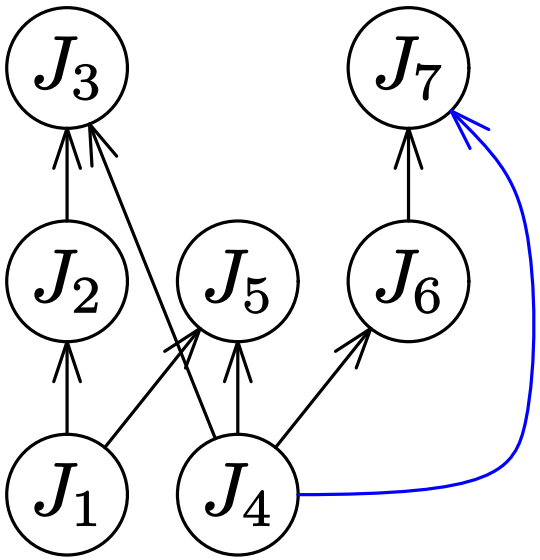
先行制約 = 閉路を持たない有向グラフ

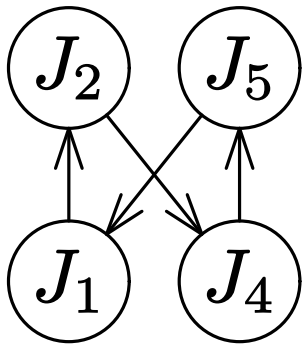




閉路の例

先行制約 = 閉路を持たない有向グラフ





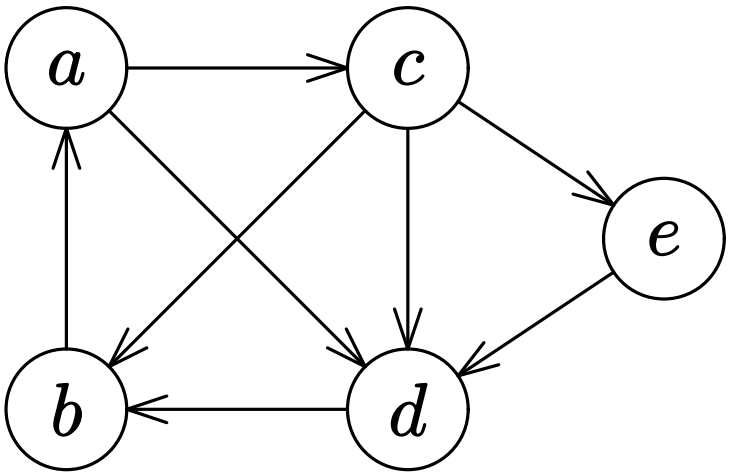
閉路の例

約束：推移辺 (transitive edge) は描かないとする

注意 (重要) (詳細は付録)

本来，先行制約は半順序 (partial order) として考えるが
この授業ではそうしない
閉路を持たない有向グラフ (で推移辺を持たないもの) は
半順序集合のハッセ図 (Hasse diagram) と同じもの

有向グラフ

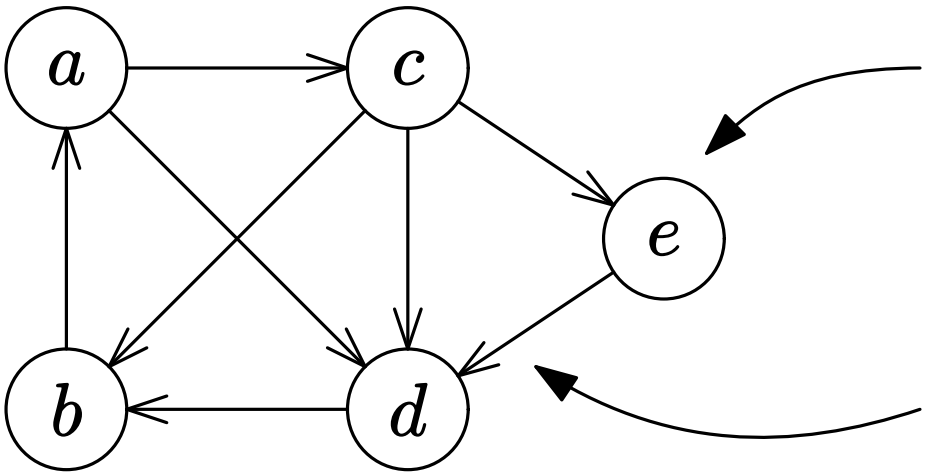


有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$

$$\mathcal{V} = \{a, b, c, d, e\}$$

$$\mathcal{A} = \{(a, c), (a, d), (b, a), (c, b), (c, d), (c, e), (d, b), (e, d)\}$$

有向グラフ



頂点 (点，節点)

vertex

辺 (弧，枝)

edge, arc

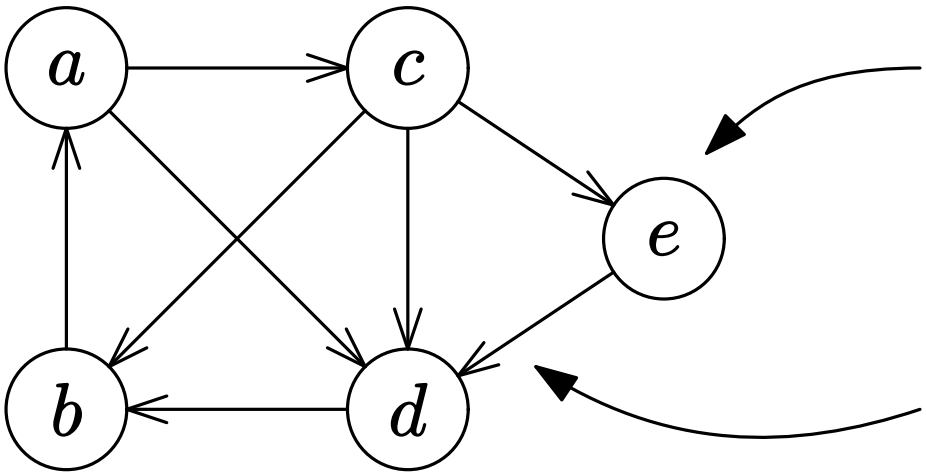
有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$

$\mathcal{V} = \{a, b, c, d, e\}$　　$\mathcal{P}$ の頂点集合

$\mathcal{A} = \{(a, c), (a, d), (b, a), (c, b), (c, d), (c, e), (d, b), (e, d)\}$

$\mathcal{P}$ の辺集合

有向グラフ



頂点 (点，節点)

vertex

辺 (弧，枝)

edge, arc

有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$

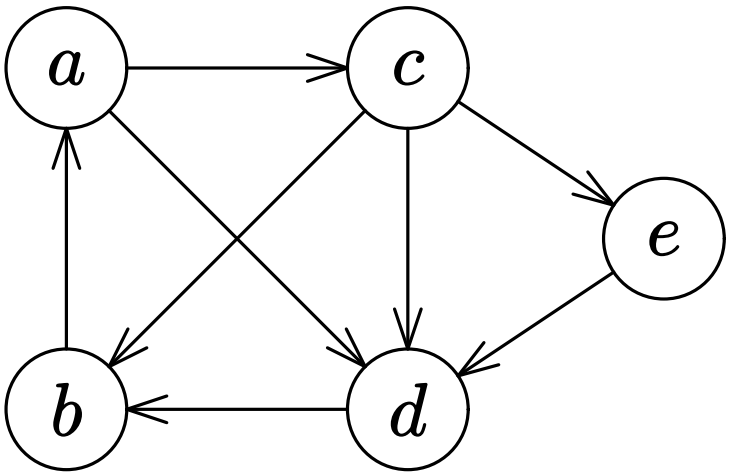
$\mathcal{V} = \{a, b, c, d, e\}$　　$\mathcal{P}$ の頂点集合

$\mathcal{A} = \{(a, c), (a, d), (b, a), (c, b), (c, d), (c, e), (d, b), (e, d)\}$

$\mathcal{P}$ の辺集合

この授業では，今後「$a \to c$」とも書く

出次数と入次数



a の出次数 (out-degree) $= 2$

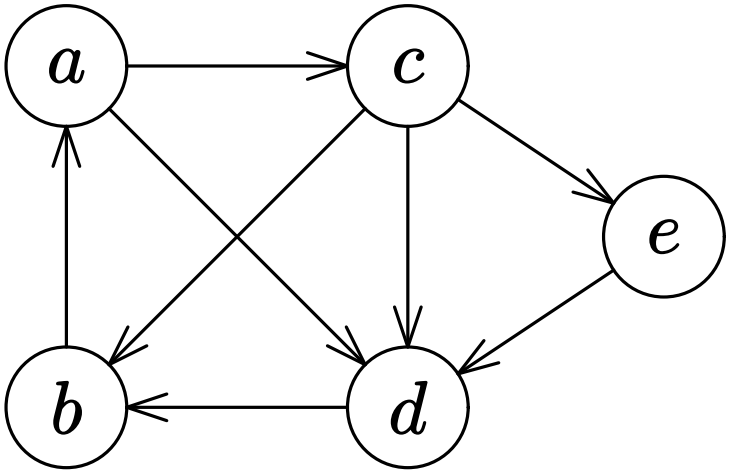
a の入次数 (in-degree) $= 1$

定義：出次数と入次数

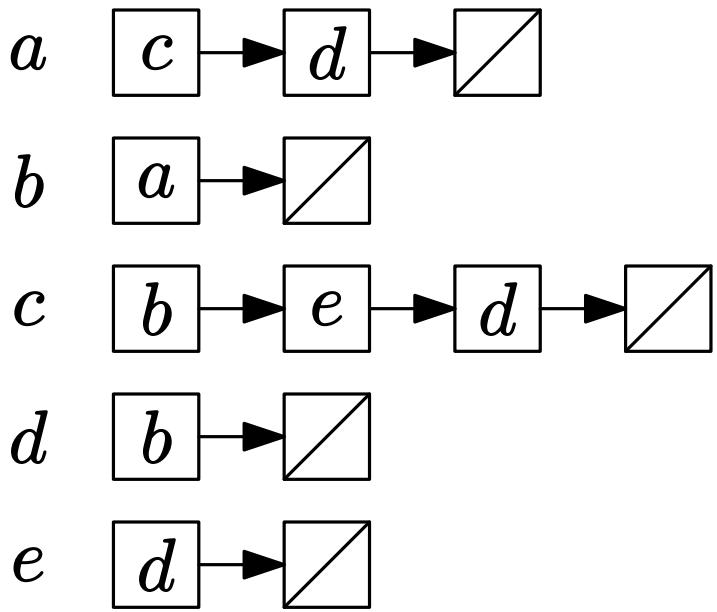
有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$ において，

- 頂点 $v \in \mathcal{V}$ の **出次数** とは，v から出る辺の総数
- 頂点 $v \in \mathcal{V}$ の **入次数** とは，v に入る辺の総数

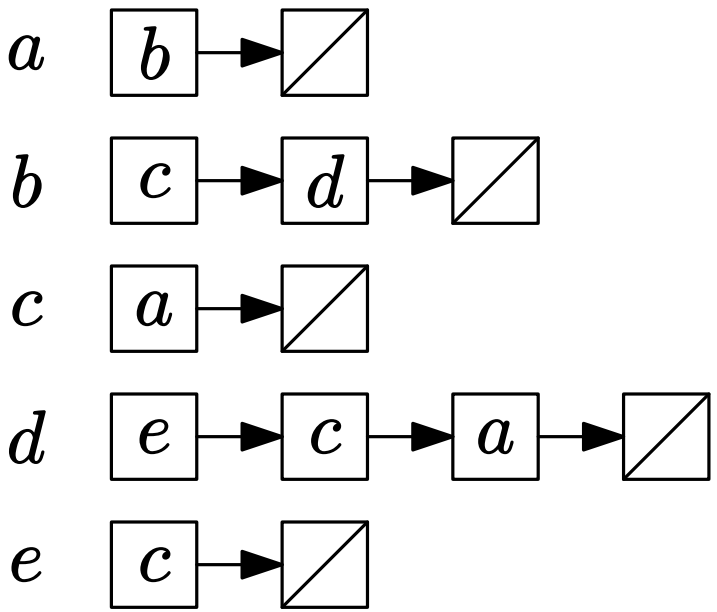
有向グラフの表現：隣接リスト



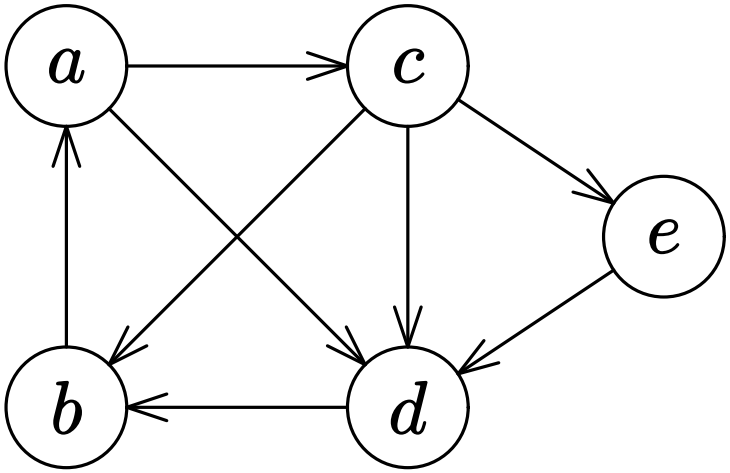
出る辺のリスト



入る辺のリスト



有向グラフの表現：隣接リスト

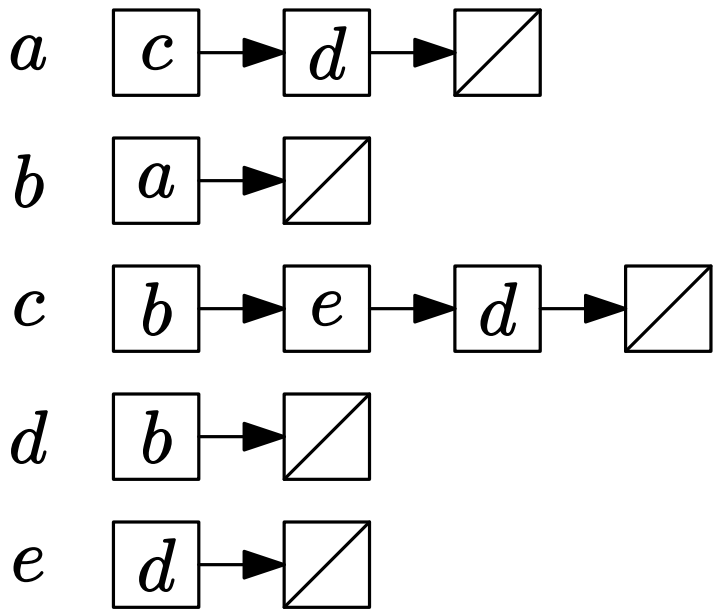


有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$

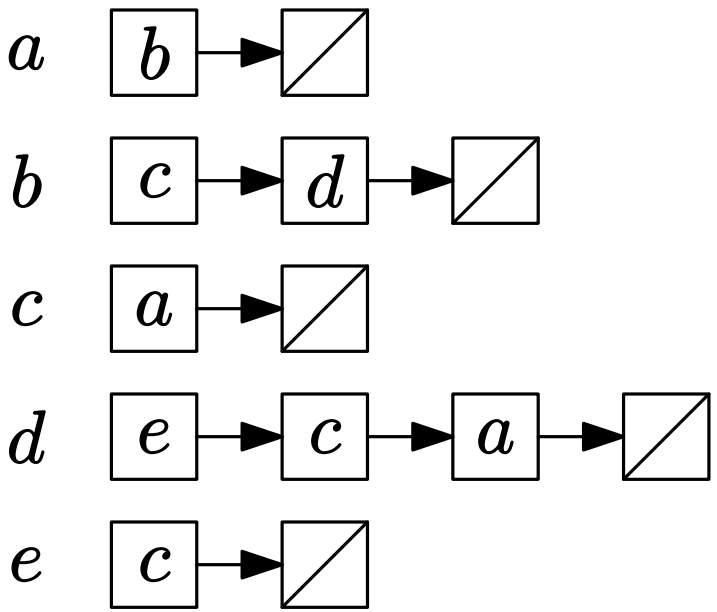
$\rightsquigarrow$ 隣接リストの大きさ

$= O((|\mathcal{V}| + |\mathcal{A}|) \log |\mathcal{V}|)$

出る辺のリスト



入る辺のリスト

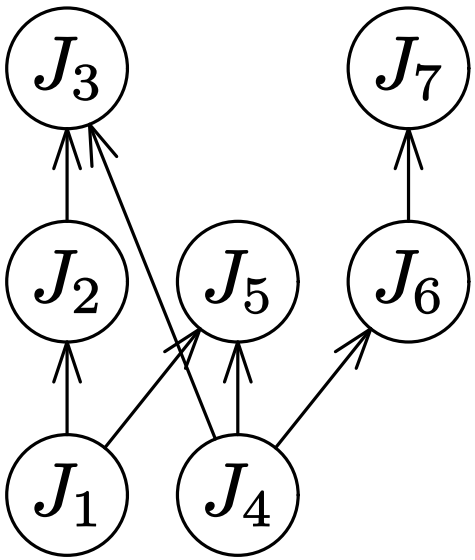


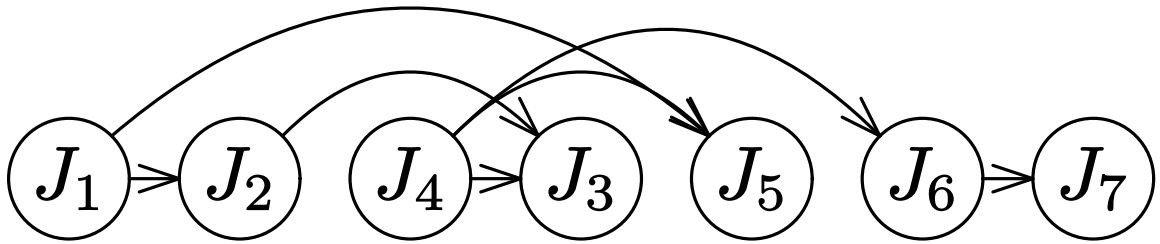
トポロジカル順序

閉路を持たない有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$, $|\mathcal{V}| = n$

定義：トポロジカル順序 (topological order)

$\mathcal{V}$ の **トポロジカル順序** とは,
頂点の並び $v_1, v_2, \ldots, v_n$ で,
$v_i \to v_j$ ならば, $i < j$ を満たすこと





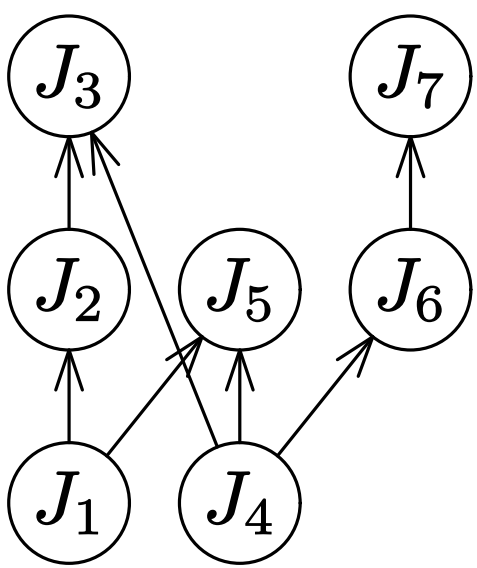
半順序では **線形拡大** (linear extension) と呼ばれる

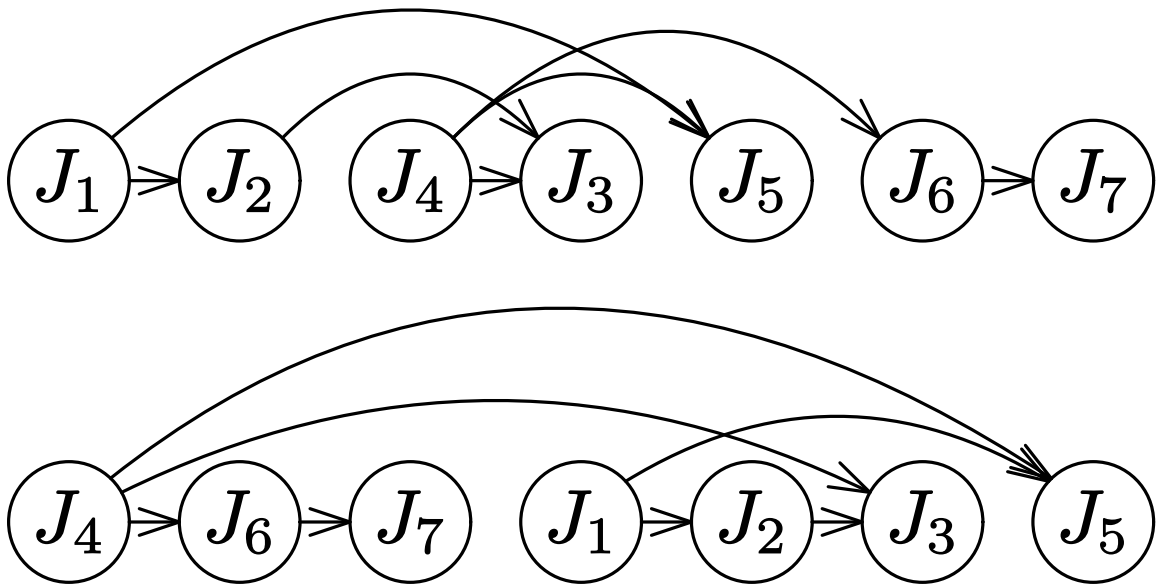
トポロジカル順序

閉路を持たない有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$, $|\mathcal{V}| = n$

定義：トポロジカル順序 (topological order)

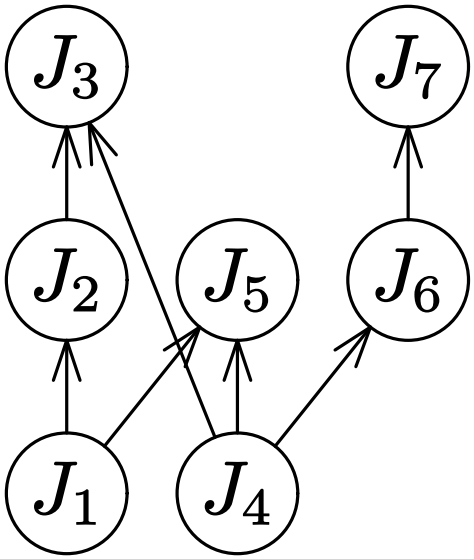
$\mathcal{V}$ の **トポロジカル順序** とは,
頂点の並び $v_1, v_2, \ldots, v_n$ で,
$v_i \to v_j$ ならば, $i < j$ を満たすこと



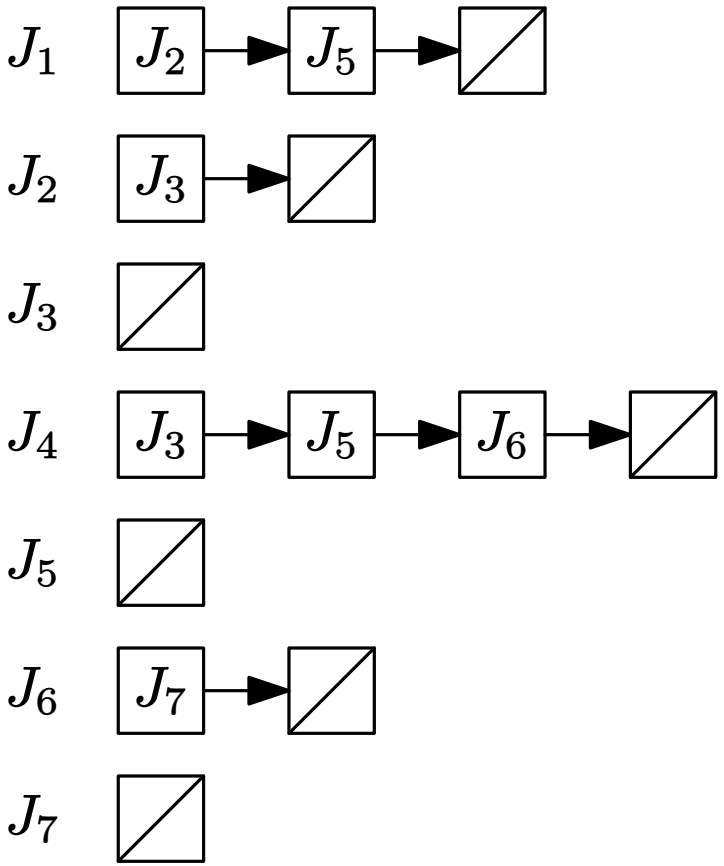


半順序では **線形拡大** (linear extension) と呼ばれる

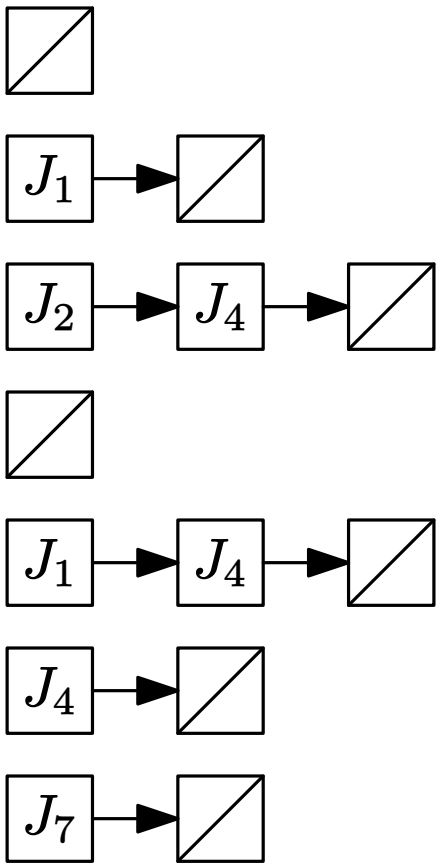
トポロジカル順序：アルゴリズムの例



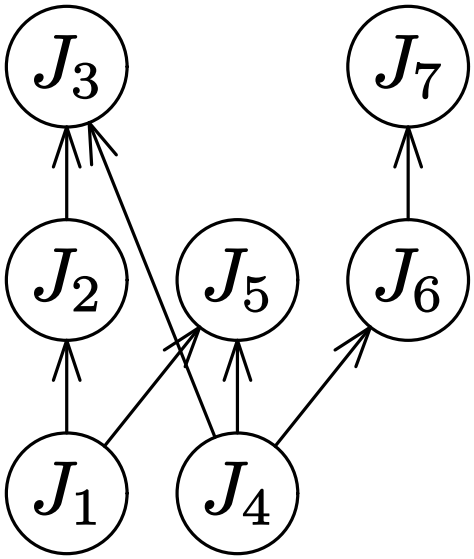
出る辺のリスト



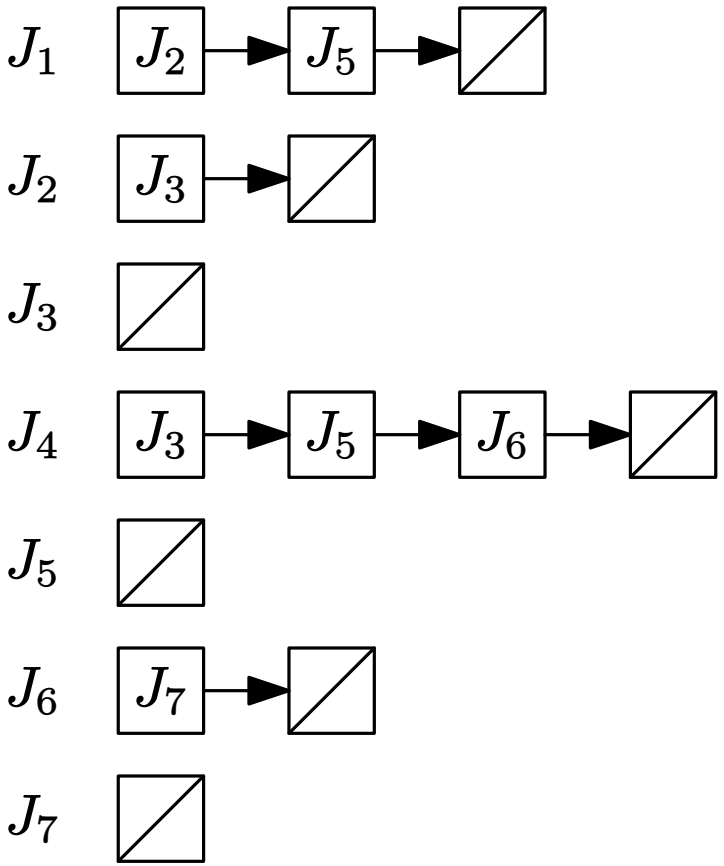
入る辺のリスト



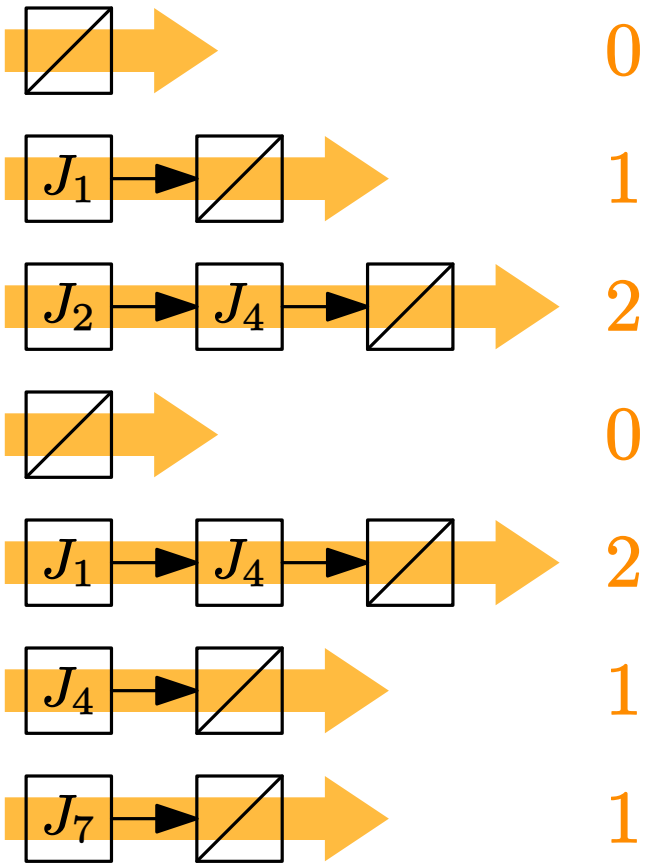
トポロジカル順序：アルゴリズムの例



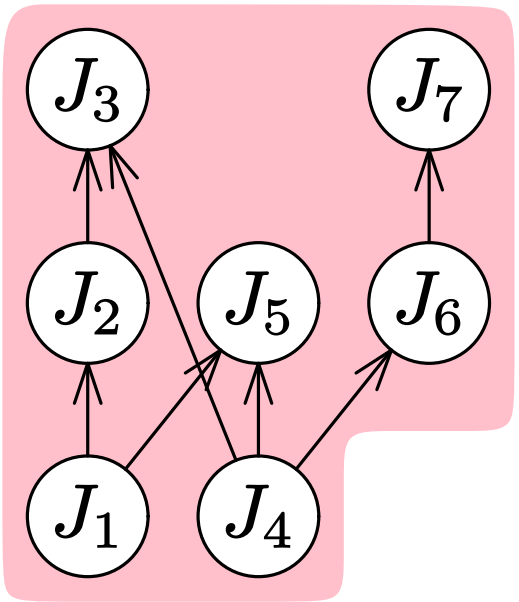
出る辺のリスト



入る辺のリスト



トポロジカル順序：アルゴリズムの例

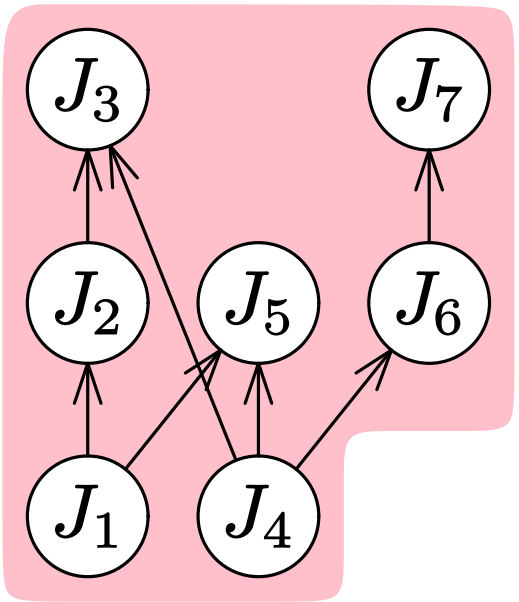


	出る辺のリスト	入次数
J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	1
J_3	／	2
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	2
J_6	J_7 → ／	1
J_7	／	1

S :

L :

トポロジカル順序：アルゴリズムの例

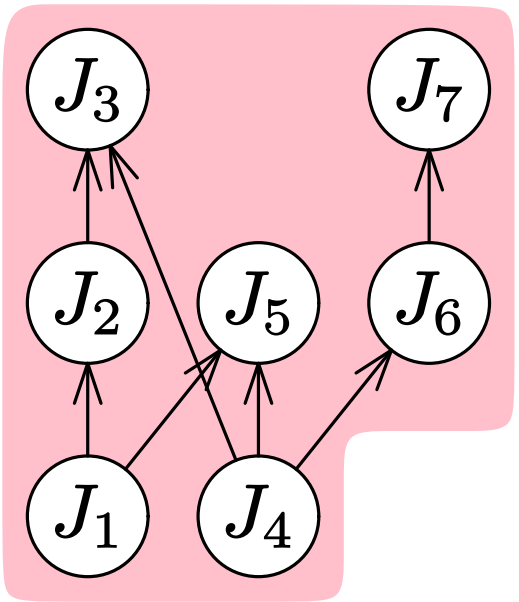


	出る辺のリスト	入次数
J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	1
J_3	／	2
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	2
J_6	J_7 → ／	1
J_7	／	1

S : J_1 J_4

L :

トポロジカル順序：アルゴリズムの例

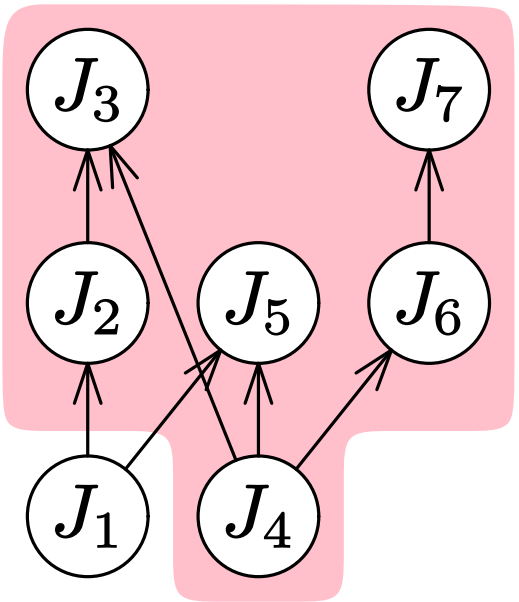


	出る辺のリスト	入次数
J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	1
J_3	／	2
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	2
J_6	J_7 → ／	1
J_7	／	1

S : ~~J_1~~ J_4

L : J_1

トポロジカル順序：アルゴリズムの例

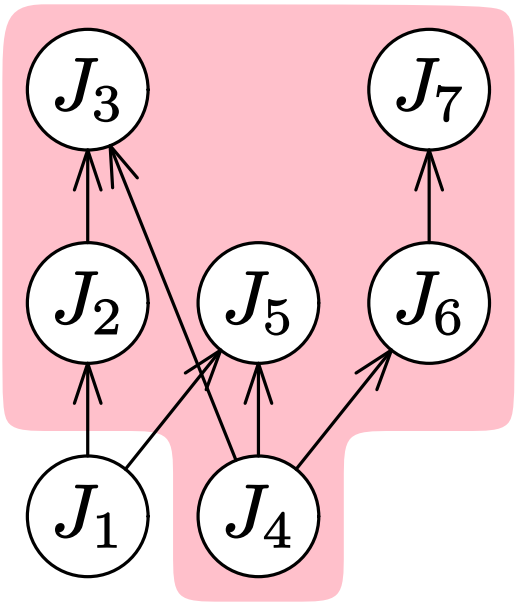


	出る辺のリスト	入次数	
J_1	J_2 → J_5 → ／	0	
J_2	J_3 → ／	~~1~~	0
J_3	／	2	
J_4	J_3 → J_5 → J_6 → ／	0	
J_5	／	~~2~~	1
J_6	J_7 → ／	1	
J_7	／	1	

S : ~~J_1~~ J_4 J_2

L : J_1

トポロジカル順序：アルゴリズムの例

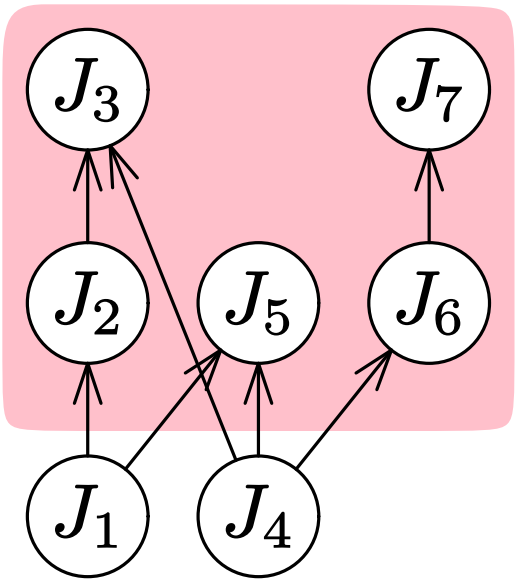


	出る辺のリスト	入次数	
J_1	J_2 → J_5 → ／	0	
J_2	J_3 → ／	~~1~~	0
J_3	／	2	
J_4	J_3 → J_5 → J_6 → ／	0	
J_5	／	~~2~~	1
J_6	J_7 → ／	1	
J_7	／	1	

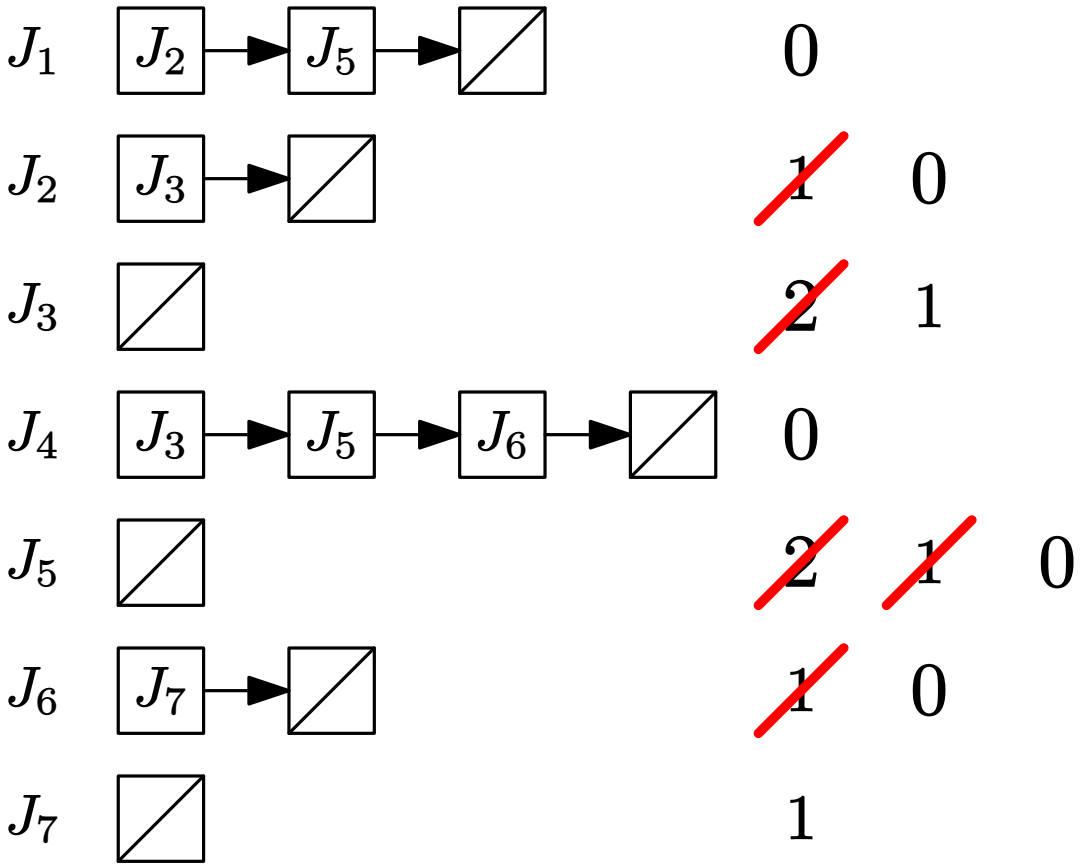
S : ~~J_1~~ ~~J_4~~ J_2

L : J_1 J_4

トポロジカル順序：アルゴリズムの例



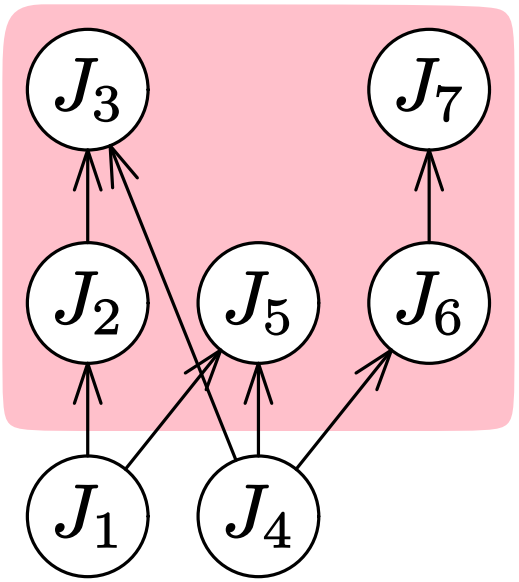
出る辺のリスト　　入次数



S : ~~J_1~~ ~~J_4~~ J_2 J_5 J_6

L : J_1 J_4

トポロジカル順序：アルゴリズムの例



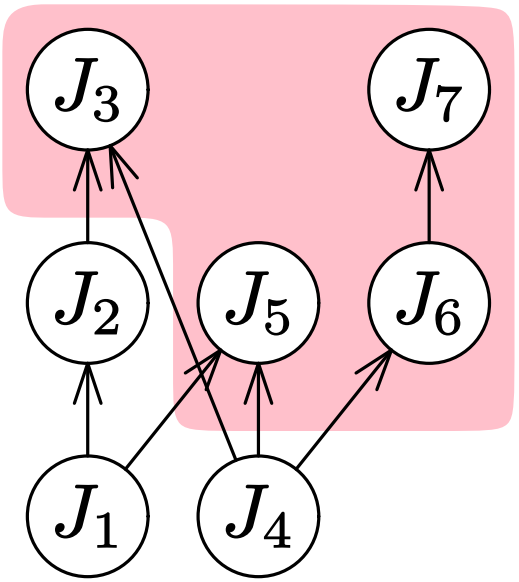
出る辺のリスト　　入次数

	出る辺のリスト	入次数
J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	~~1~~ 0
J_3	／	~~2~~ 1
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	~~2~~ ~~1~~ 0
J_6	J_7 → ／	~~1~~ 0
J_7	／	1

S : ~~J_1~~ ~~J_4~~ ~~J_2~~ J_5 J_6

L : J_1 J_4 J_2

トポロジカル順序：アルゴリズムの例



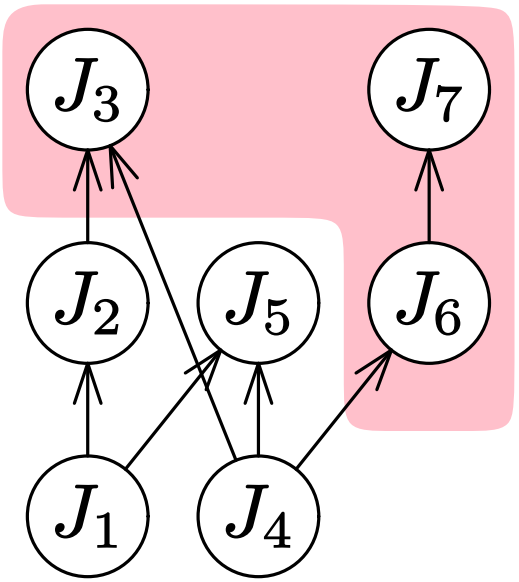
出る辺のリスト　　入次数

	出る辺のリスト	入次数
J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	~~1~~ 0
J_3	／	~~2~~ ~~1~~ 0
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	~~2~~ ~~1~~ 0
J_6	J_7 → ／	~~1~~ 0
J_7	／	1

S : ~~J_1~~ ~~J_4~~ ~~J_2~~ J_5 J_6 J_3

L : J_1 J_4 J_2

トポロジカル順序：アルゴリズムの例



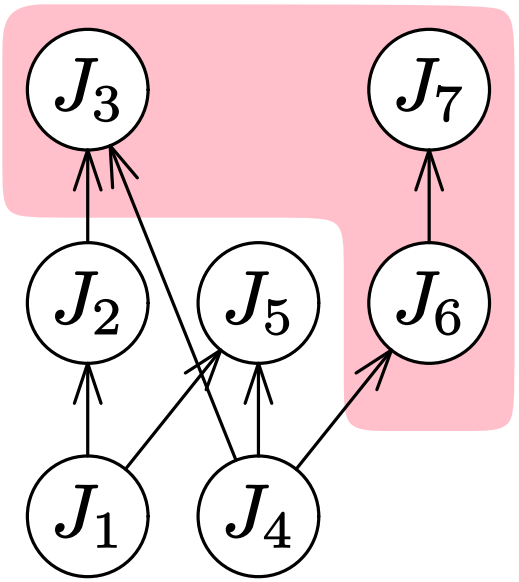
出る辺のリスト　　入次数

	出る辺のリスト	入次数
J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	~~1~~ 0
J_3	／	~~2~~ ~~1~~ 0
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	~~2~~ ~~1~~ 0
J_6	J_7 → ／	~~1~~ 0
J_7	／	1

S : ~~J_1~~ ~~J_4~~ ~~J_2~~ ~~J_5~~ J_6 J_3

L : J_1 J_4 J_2 J_5

トポロジカル順序：アルゴリズムの例



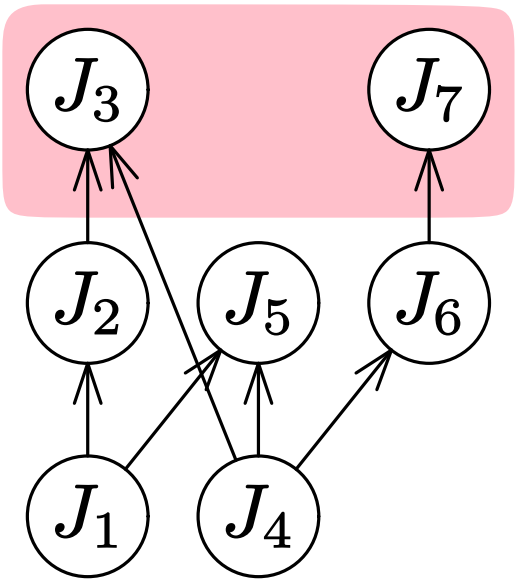
出る辺のリスト

		入次数
J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	~~1~~ 0
J_3	／	~~2~~ ~~1~~ 0
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	~~2~~ ~~1~~ 0
J_6	J_7 → ／	~~1~~ 0
J_7	／	1

S : ~~J_1~~ ~~J_4~~ ~~J_2~~ ~~J_5~~ ~~J_6~~ J_3

L : J_1 J_4 J_2 J_5 J_6

トポロジカル順序：アルゴリズムの例



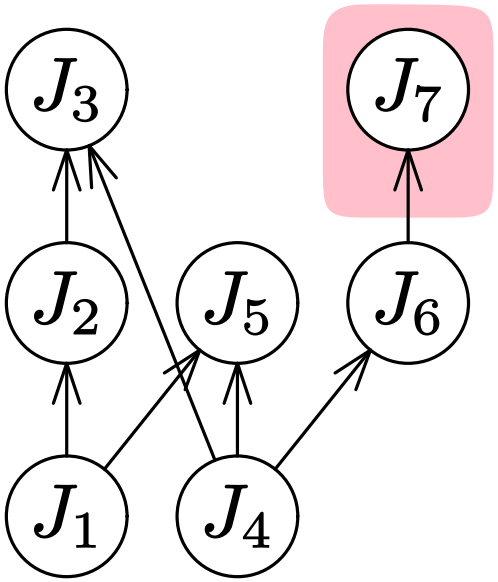
出る辺のリスト　　入次数

	出る辺のリスト	入次数
J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	~~1~~ 0
J_3	／	~~2~~ ~~1~~ 0
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	~~2~~ ~~1~~ 0
J_6	J_7 → ／	~~1~~ 0
J_7	／	~~1~~ 0

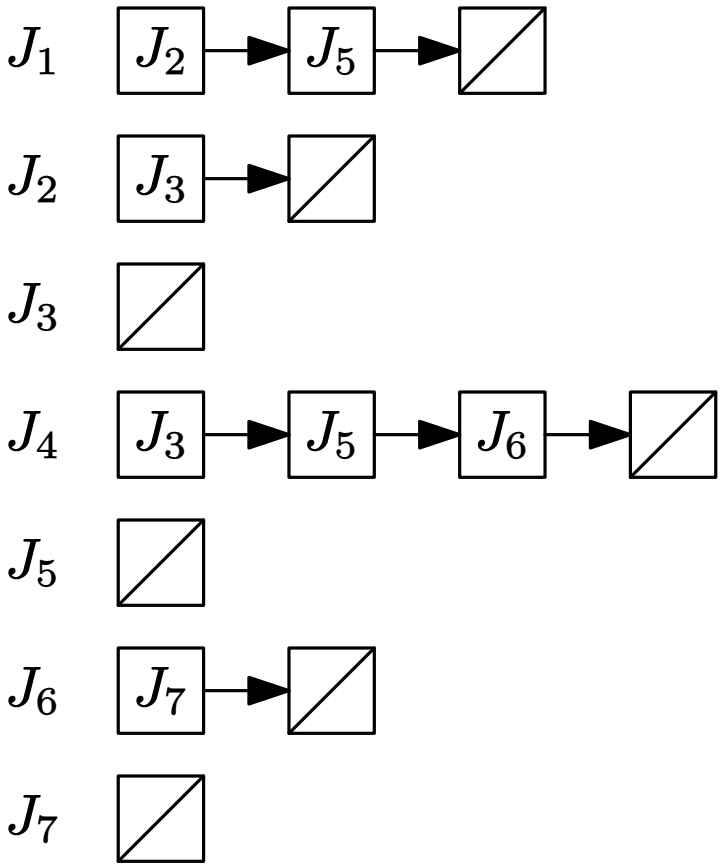
S : ~~J_1~~ ~~J_4~~ ~~J_2~~ ~~J_5~~ ~~J_6~~ J_3 J_7

L : J_1 J_4 J_2 J_5 J_6

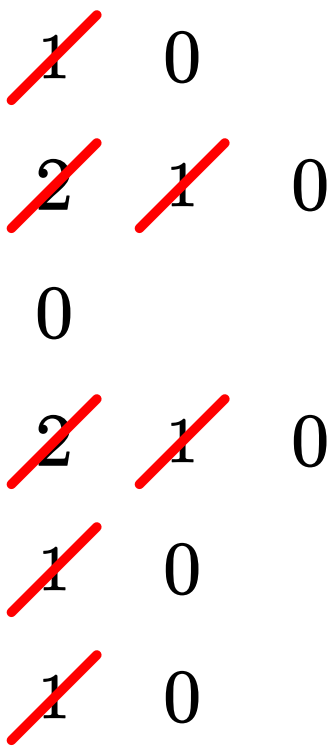
トポロジカル順序：アルゴリズムの例



出る辺のリスト



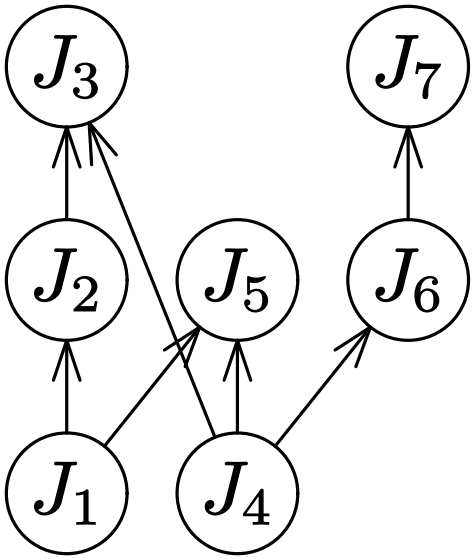
入次数



S : ~~J_1~~ ~~J_4~~ ~~J_2~~ ~~J_5~~ ~~J_6~~ ~~J_3~~ J_7

L : J_1 J_4 J_2 J_5 J_6 J_3

トポロジカル順序：アルゴリズムの例



出る辺のリスト　　入次数

J_1	J_2 → J_5 → ／	0
J_2	J_3 → ／	~~1~~ 0
J_3	／	~~2~~ ~~1~~ 0
J_4	J_3 → J_5 → J_6 → ／	0
J_5	／	~~2~~ ~~1~~ 0
J_6	J_7 → ／	~~1~~ 0
J_7	／	~~1~~ 0

S : ~~J_1~~ ~~J_4~~ ~~J_2~~ ~~J_5~~ ~~J_6~~ ~~J_3~~ ~~J_7~~

L : J_1 J_4 J_2 J_5 J_6 J_3 J_7

トポロジカル順序：アルゴリズム (1)

入力：閉路を持たない有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$

アルゴリズム：トポロジカル・ソート (Kahn '62)

1. 各頂点の入次数を計算する
2. $S =$ {入次数 0 の頂点}，$L =$ 空リスト とする
3. $S \neq \emptyset$ である限り，次を反復する
   (a) S から頂点を 1 つ取り出し，v とする
   (b) L の末尾に v を追加する
   (c) $v \to u$ のとき，u の入次数を 1 だけ減らす
4. L を出力する

計算量：$O(|\mathcal{V}| + |\mathcal{A}|)$

以後，$n = |\mathcal{V}| = |\mathcal{J}|, h = |\mathcal{A}|$

トポロジカル順序：アルゴリズム (2)

入力：閉路を持たない有向グラフ $\mathcal{P} = (\mathcal{V}, \mathcal{A})$

アルゴリズム：トポロジカル・ソート

1. 各頂点の出次数を計算する
2. $S = \{$出次数 0 の頂点$\}$，$L =$ 空リスト とする
3. $S \neq \emptyset$ である限り，次を反復する
   (a) S から頂点を 1 つ取り出し，v とする
   (b) L の先頭に v を追加する
   (c) $u \to v$ のとき，u の出次数を 1 だけ減らす
4. L を出力する

計算量：$O(|\mathcal{V}| + |\mathcal{A}|)$

以後，$n = |\mathcal{V}| = |\mathcal{J}|, h = |\mathcal{A}|$

先行制約がない = 特殊な先行制約がある

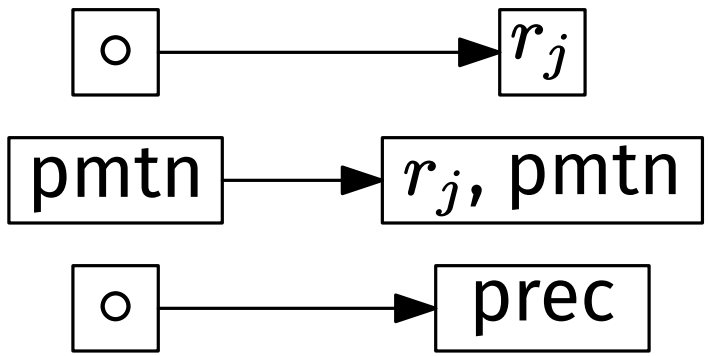
「先行制約がない」場合を表す 有向グラフ (閉路を持たない)

先行制約がない = 特殊な先行制約がある

「先行制約がない」場合を表す 有向グラフ (閉路を持たない)

計算量

β　ジョブの特性



本日の内容

1. 先行制約の基礎概念
2. **一機械スケジューリングと先行制約**
3. $P\infty \mid \mathrm{prec} \mid C_{\max}$ とクリティカル・パス法

- E. L. Lawler, Optimal sequencing for a single machine subject to precedence constraints. *Management Science* (1973) 19, pp. 544–546.
- E. L. Lawler, Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Mathematics* 2 (1978) pp. 75–90.
- J. K. Lenstra, A. H. G. Rinnooy Kan, Computational complexity of scheduling under precedence constraints. *Operations Research* 26 (1978) pp. 22–35.
- M. R. Garey, D. S. Johnson, Scheduling tasks with nonuniform deadlines on two processors. *Journal of the ACM* 23 (1976) pp. 461–467.

まとめ：一機械，先行制約あり

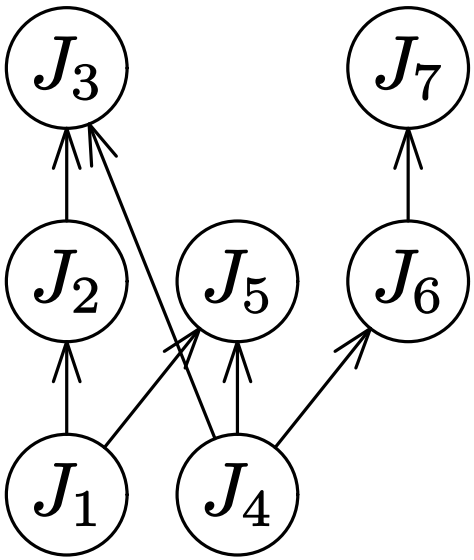
まとめ : 1 | prec | γ

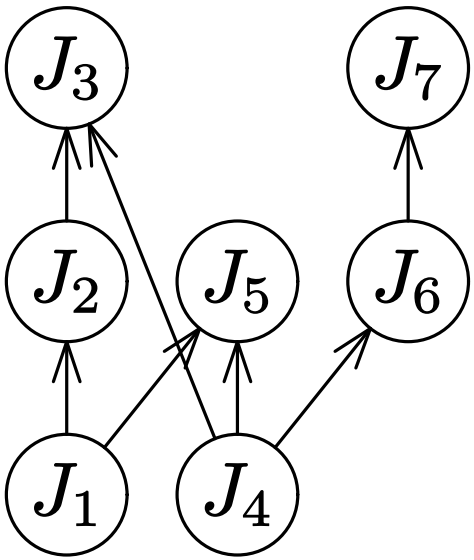
γ		計算量	(文献)
$C_{\max}$	最大完了時刻	$O(n+h)$	
$\sum C_j$	総完了時刻	強 NP 困難	(Lawler '78)
$L_{\max}$	最大納期ずれ	$O(n \log n + h)$	(Lawler '73)
$\sum T_j$	総納期遅れ	強 NP 困難	(Lenstra, Rinnooy Kan '78)
$\sum U_j$	納期遅れジョブ数	強 NP 困難	(Garey, Johnson '76)

まとめ：一機械，先行制約あり

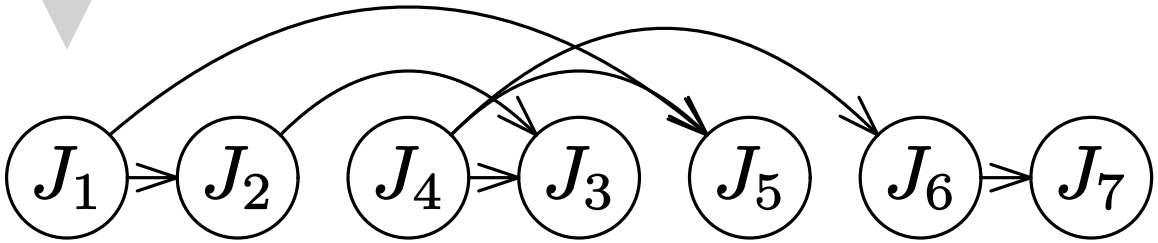
まとめ : 1 | prec | γ

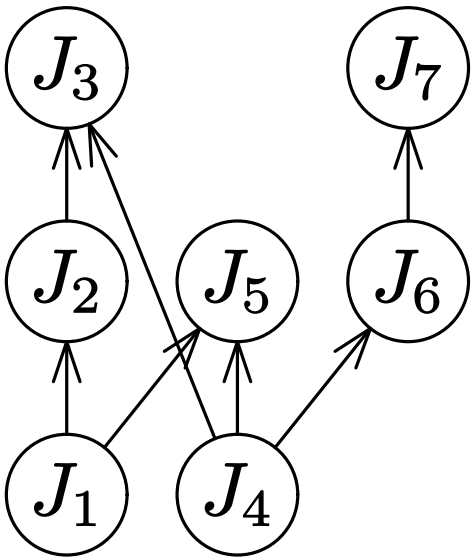
γ		計算量	(文献)
$C_{\max}$	最大完了時刻	$O(n+h)$	
$\sum C_j$	総完了時刻	強 NP 困難	(Lawler '78)
$L_{\max}$	最大納期ずれ	$O(n \log n + h)$	(Lawler '73)
$\sum T_j$	総納期遅れ	強 NP 困難	(Lenstra, Rinnooy Kan '78)
$\sum U_j$	納期遅れジョブ数	強 NP 困難	(Garey, Johnson '76)


J_3
J_7
J_2
J_5
J_6
J_1
J_4

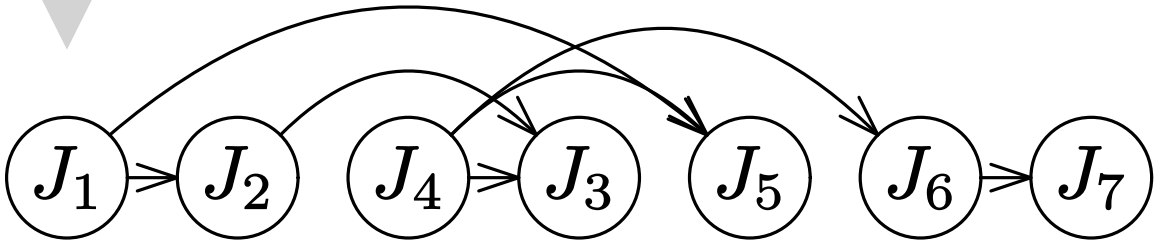


トポロジカル・ソート

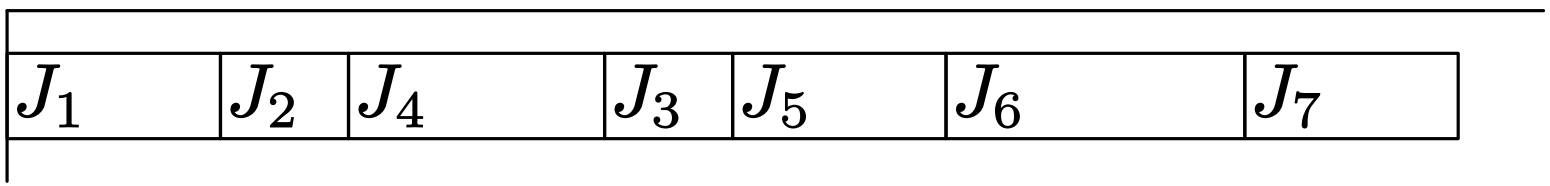




トポロジカル・ソート



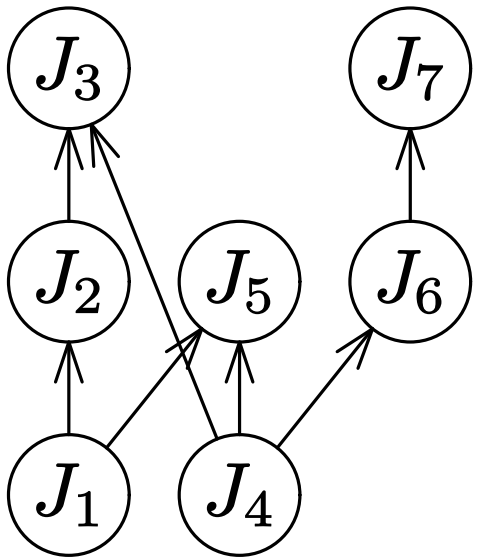
その順でジョブを並べる



1 | prec | $L_{\max}$: 例

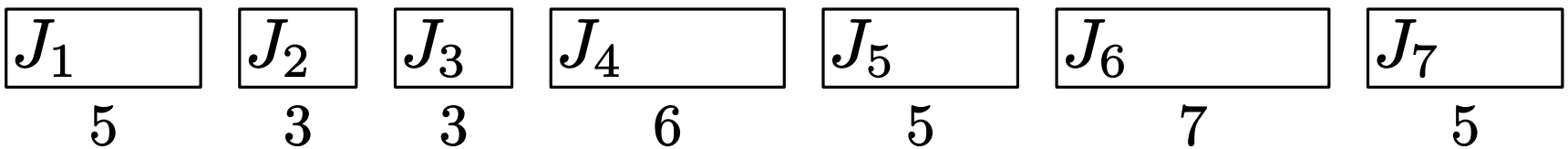
トポロジカル順序の作り方を工夫する

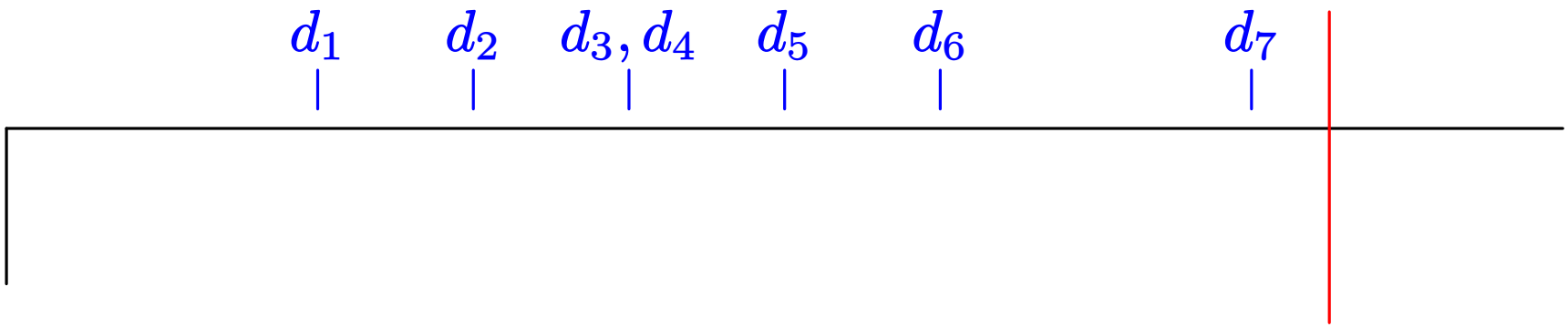
- 後ろから作る
- 出次数 $= 0$ の頂点の中から
  納期が最も遅いものを選ぶ



計算量：$O(n \log n + h)$

(S を優先度付きキューで管理する)

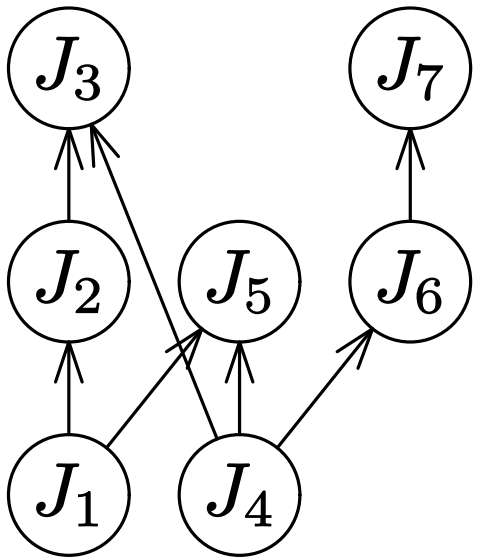




1 | prec | $L_{\max}$: 例

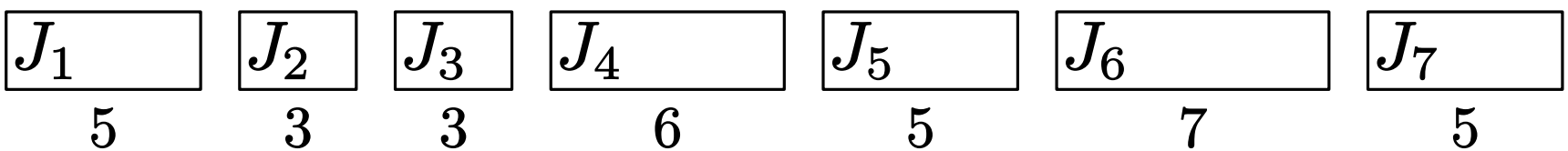
トポロジカル順序の作り方を工夫する

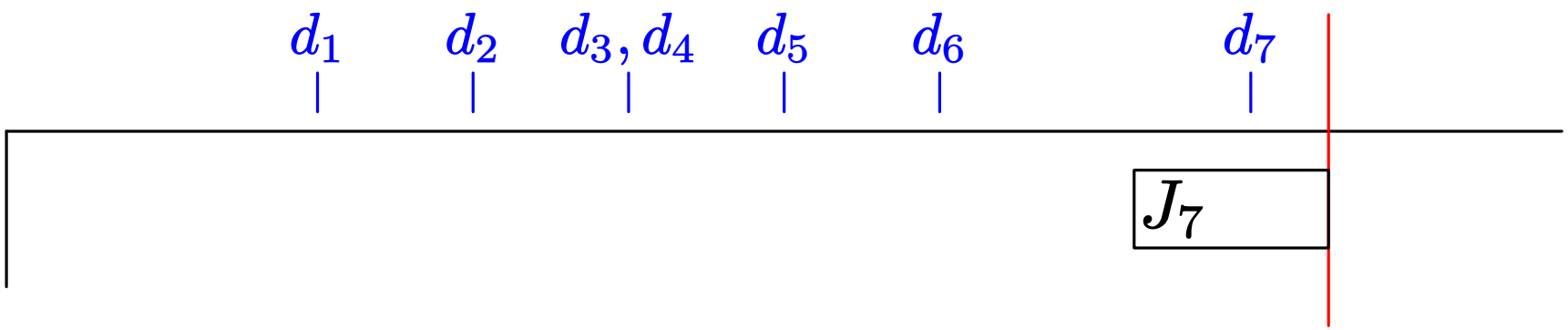
- 後ろから作る
- 出次数 $= 0$ の頂点の中から
  納期が最も遅いものを選ぶ



計算量：$O(n \log n + h)$

(S を優先度付きキューで管理する)

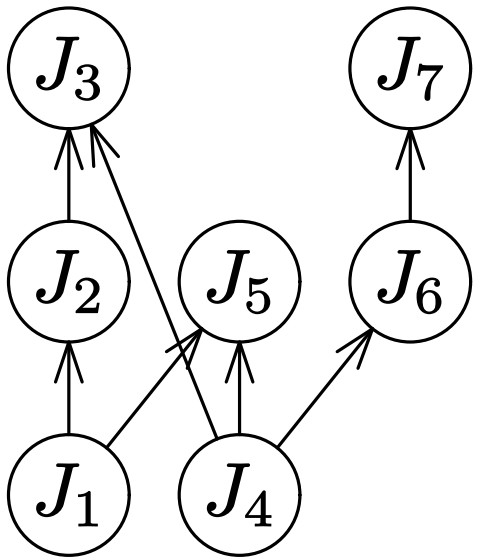




1 | prec | $L_{\max}$: 例

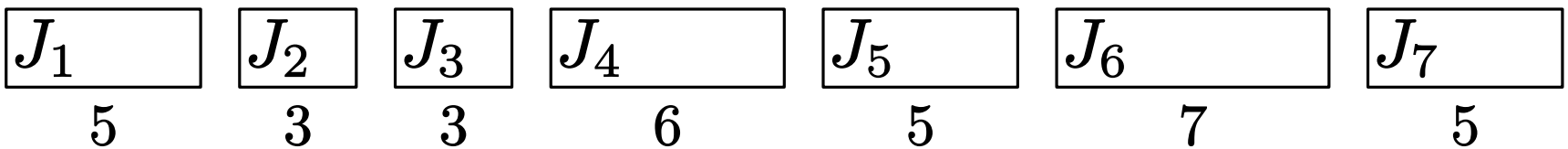
トポロジカル順序の作り方を工夫する

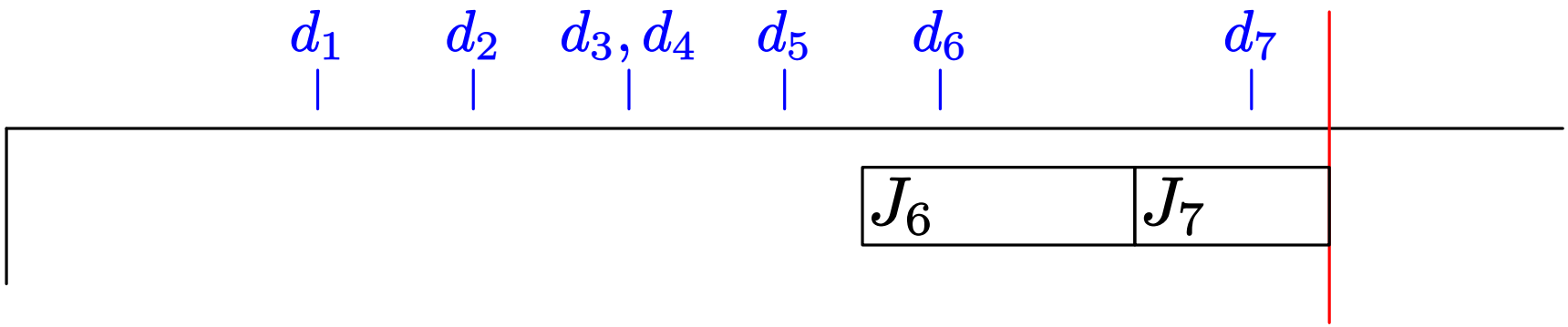
- 後ろから作る
- 出次数 $= 0$ の頂点の中から
  納期が最も遅いものを選ぶ



計算量：$O(n \log n + h)$

(S を優先度付きキューで管理する)

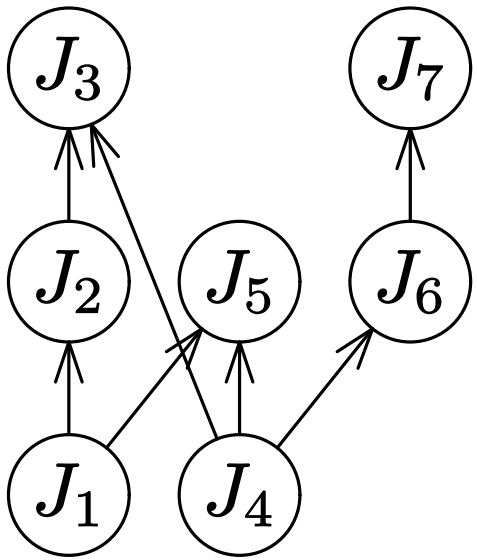




1 | prec | $L_{\max}$: 例

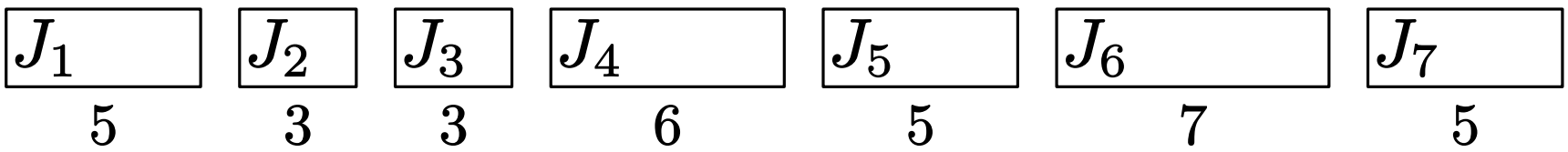
トポロジカル順序の作り方を工夫する

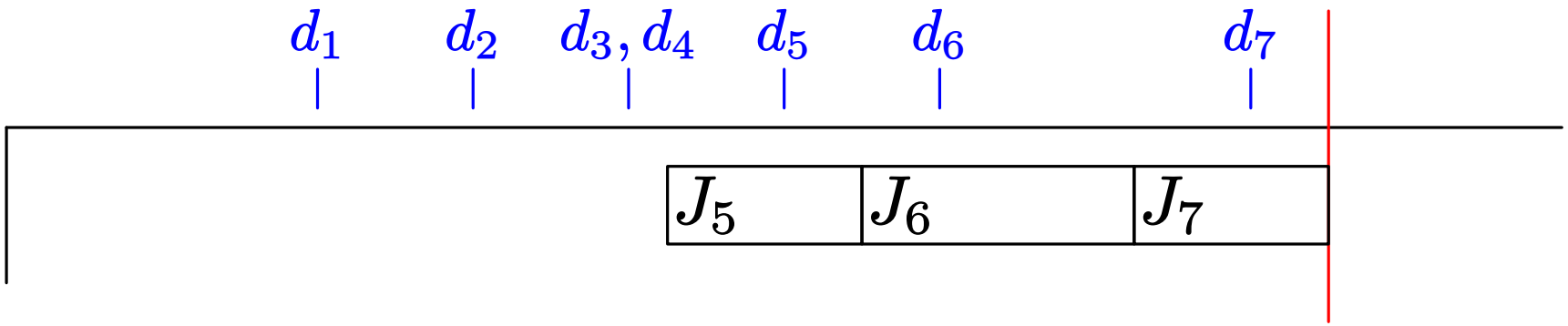
- 後ろから作る
- 出次数 $= 0$ の頂点の中から
  納期が最も遅いものを選ぶ



計算量：$O(n \log n + h)$

(S を優先度付きキューで管理する)

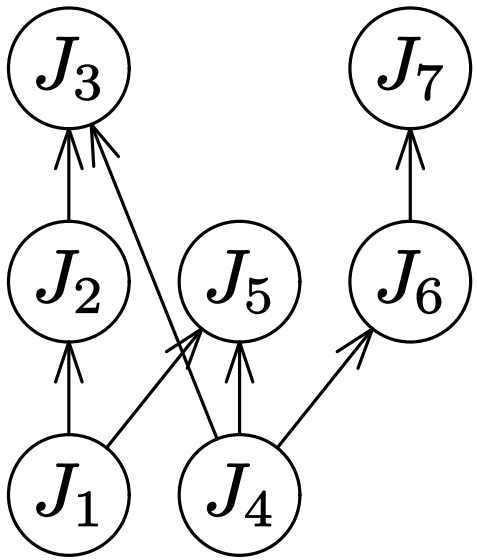




1 | prec | $L_{\max}$: 例

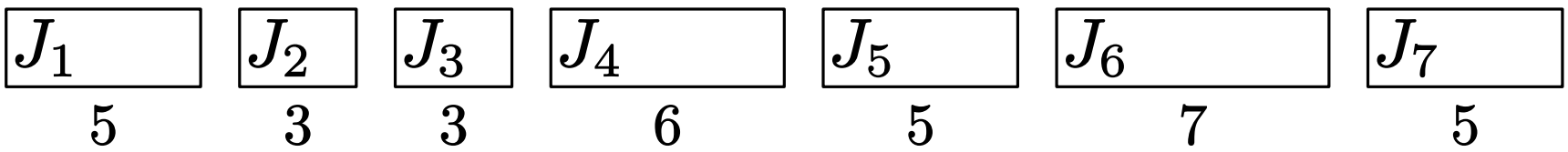
トポロジカル順序の作り方を工夫する

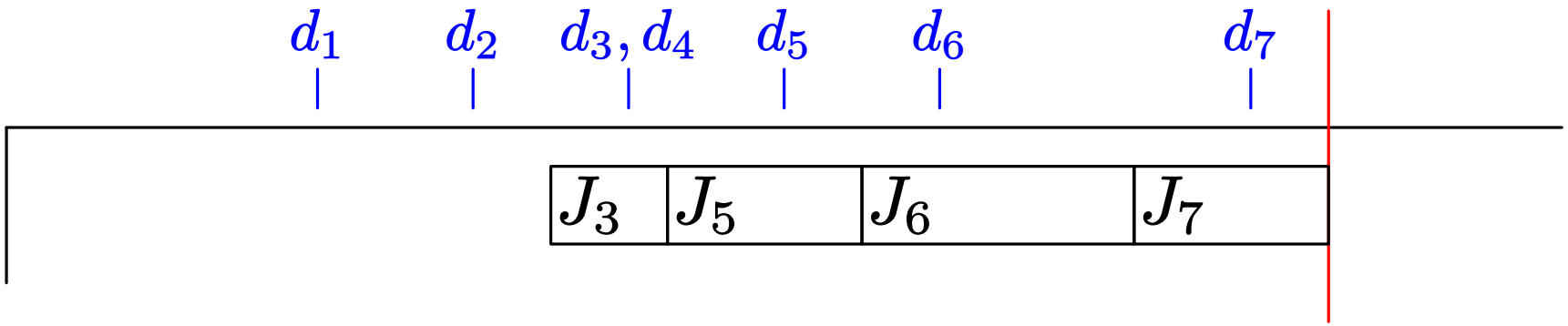
- 後ろから作る
- 出次数 $= 0$ の頂点の中から
  納期が最も遅いものを選ぶ



計算量：$O(n \log n + h)$

(S を優先度付きキューで管理する)

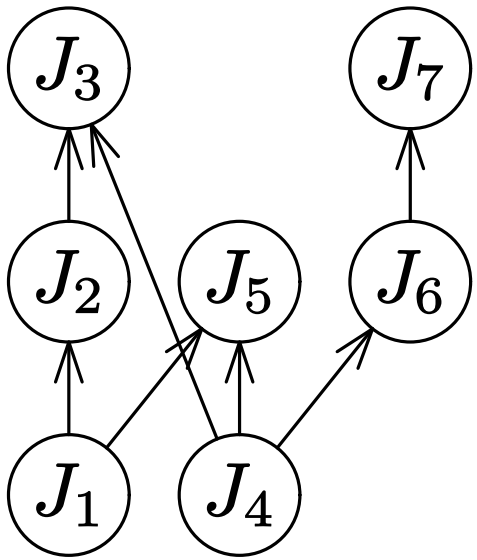




1 | prec | $L_{\max}$: 例

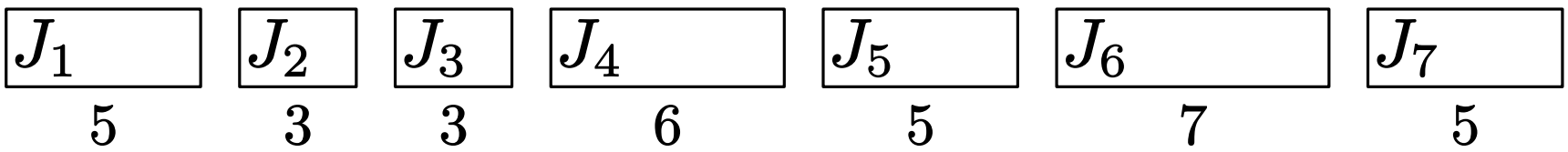
トポロジカル順序の作り方を工夫する

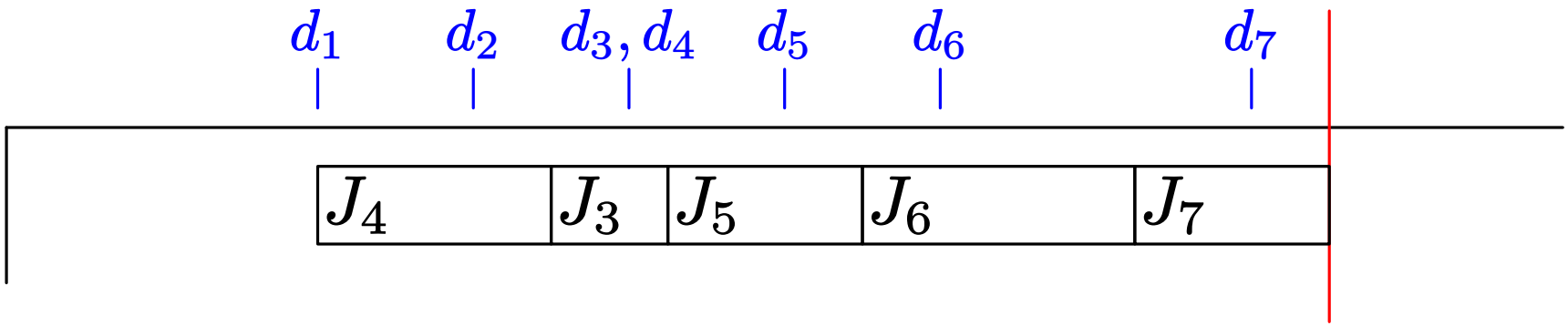
- 後ろから作る
- 出次数 $= 0$ の頂点の中から
  納期が最も遅いものを選ぶ



計算量：$O(n \log n + h)$

(S を優先度付きキューで管理する)

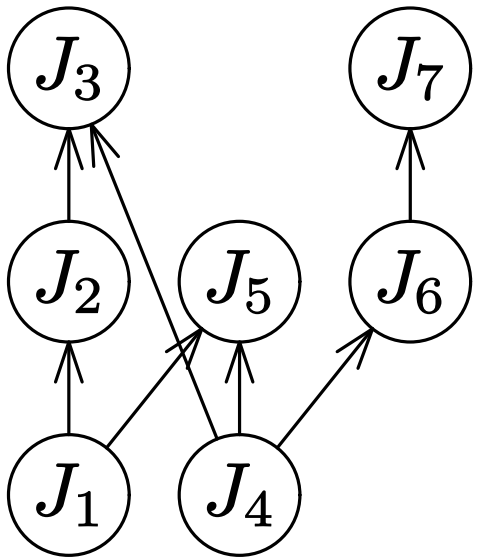




1 | prec | $L_{\max}$: 例

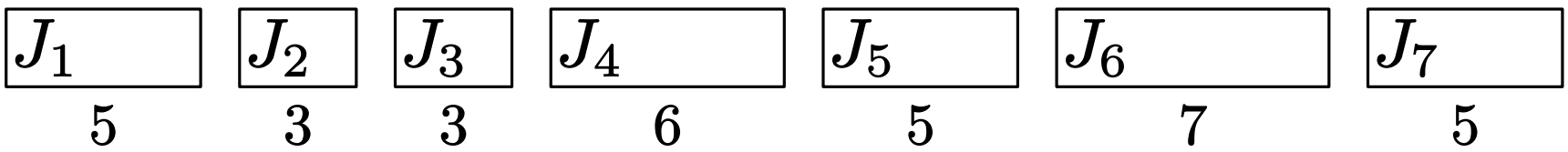
トポロジカル順序の作り方を工夫する

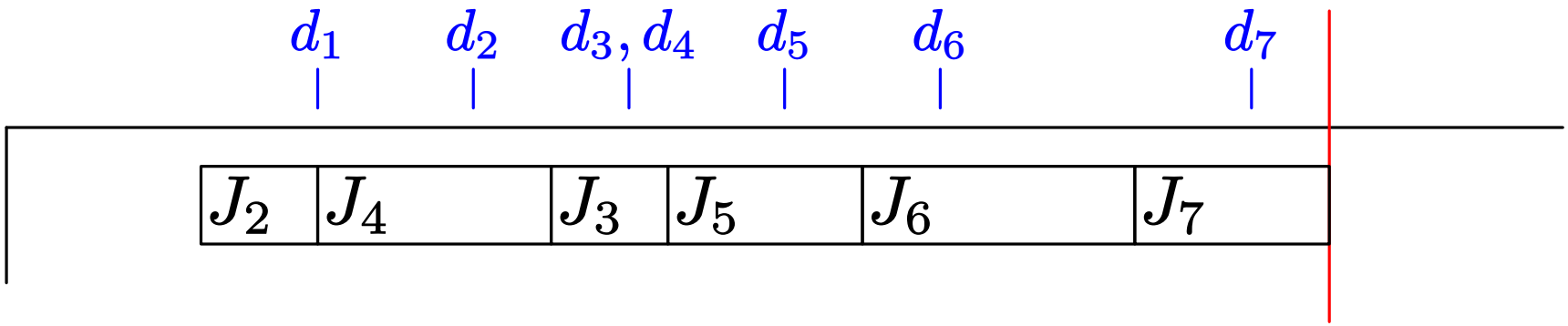
- 後ろから作る
- 出次数 $= 0$ の頂点の中から
  納期が最も遅いものを選ぶ



計算量：$O(n \log n + h)$

(S を優先度付きキューで管理する)

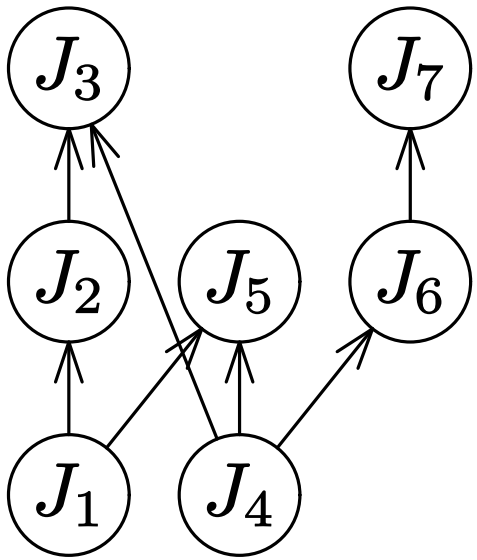




1 | prec | $L_{\max}$: 例

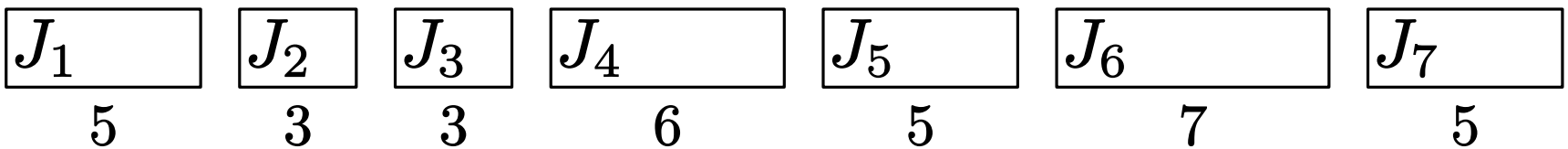
トポロジカル順序の作り方を工夫する

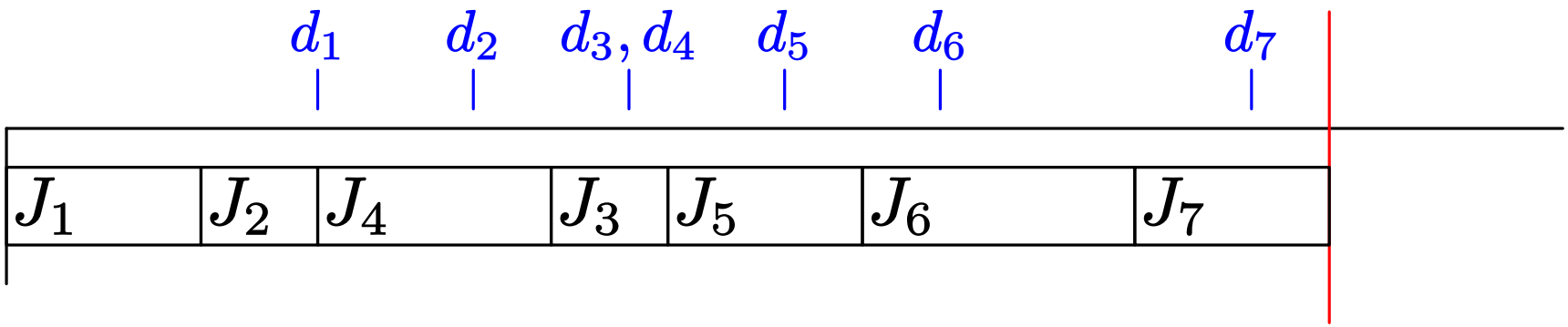
- 後ろから作る
- 出次数 $= 0$ の頂点の中から
  納期が最も遅いものを選ぶ



計算量：$O(n \log n + h)$

(S を優先度付きキューで管理する)





$1 \mid \mathrm{prec} \mid L_{\max}$：アルゴリズム

定理 (Lawler '73)

次のアルゴリズムで $1 \mid \mathrm{prec} \mid L_{\max}$ は解ける

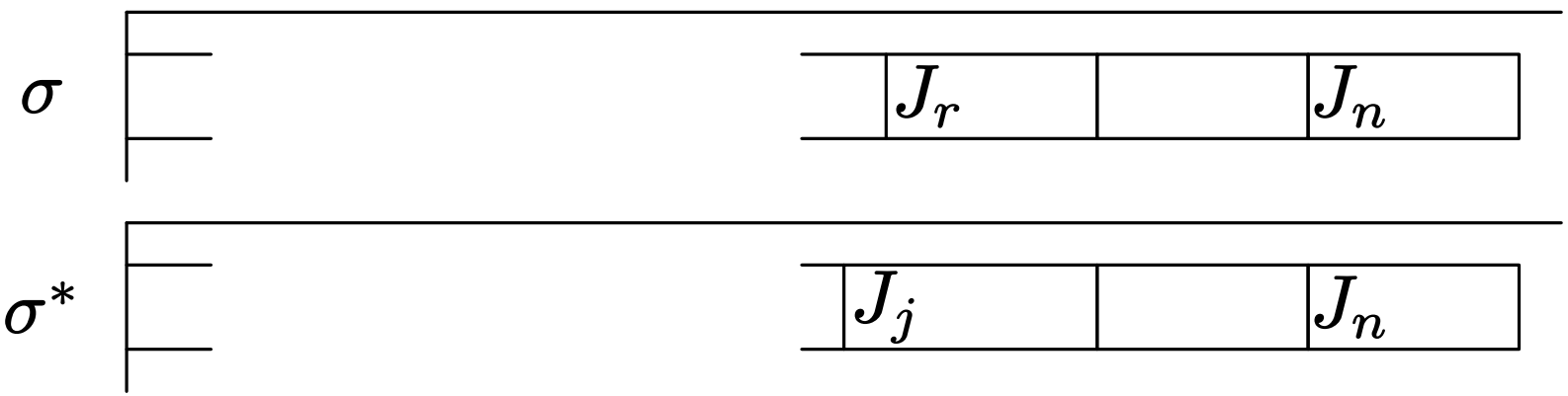
アルゴリズム：$1 \mid \mathrm{prec} \mid L_{\max}$

1. 前のページのとおり，トポロジカル順序を作る
2. その順序でジョブを処理する

計算量：$O(n \log n + h)$

1 | prec | $L_{\max}$：証明 (1)

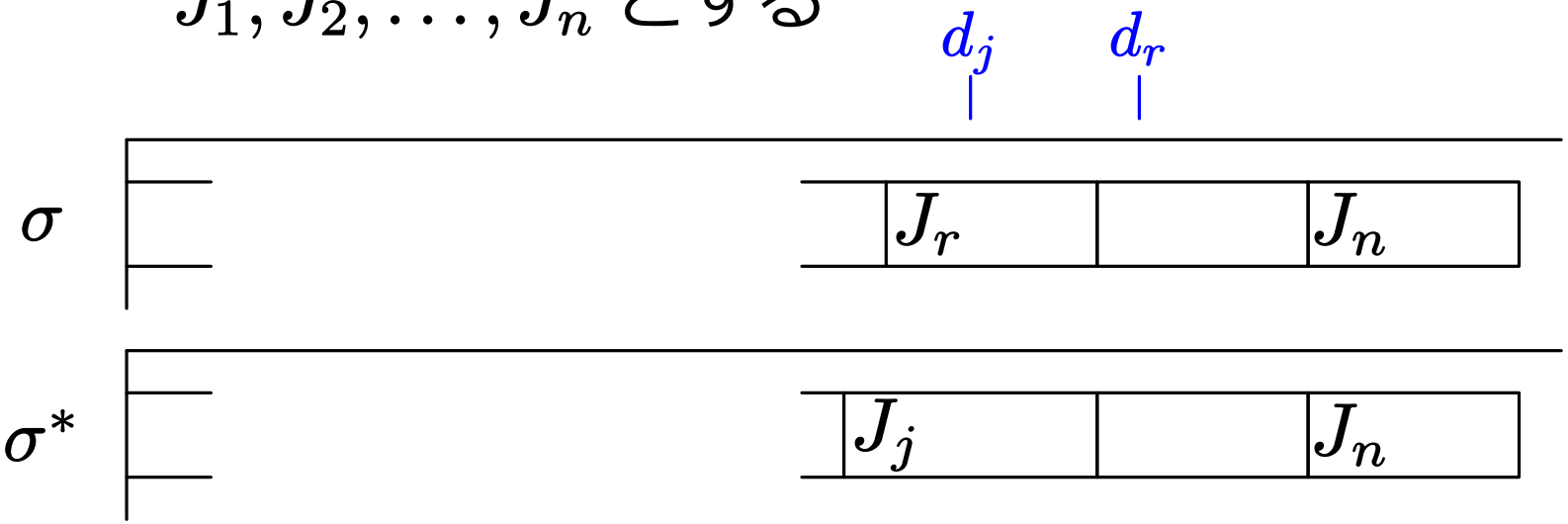
証明：アルゴリズムの作るトポロジカル順序 σ を
$J_1, J_2, \ldots, J_n$ とする



- σ^* を最適解として，完了時刻の遅い方から見て
  始めに σ と σ^* で異なるジョブに着目する
- そこにおける σ のジョブを J_r，σ^* のジョブを J_j とする

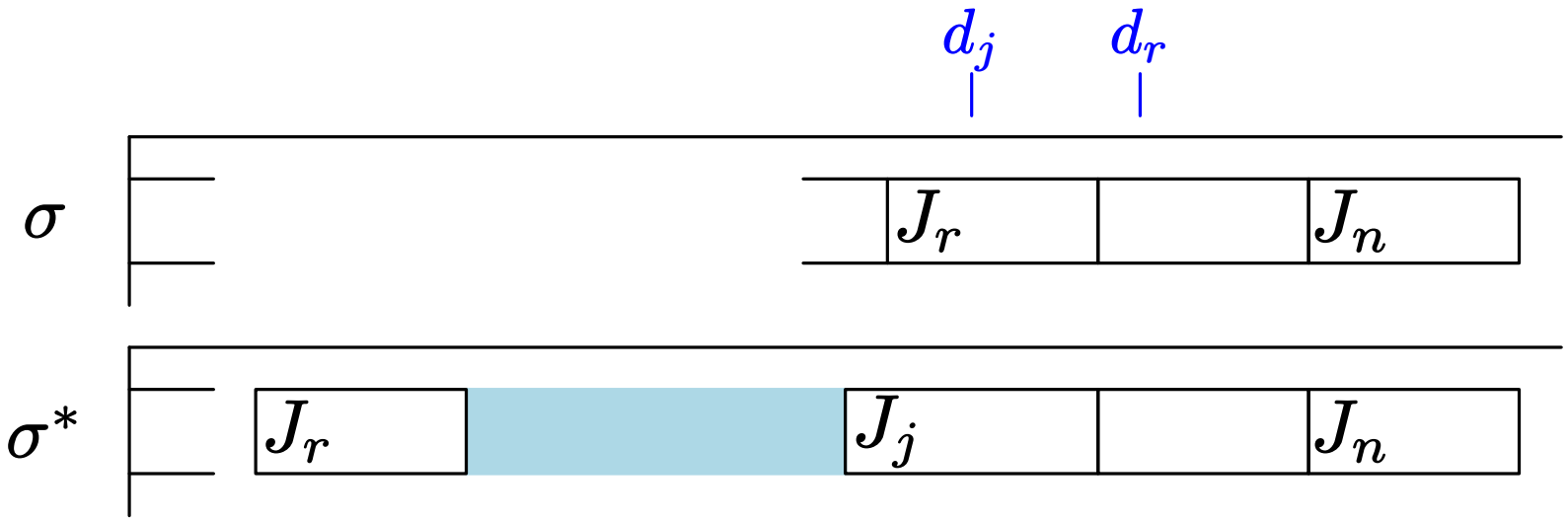
1 | prec | $L_{\max}$: 証明 (1)

証明：アルゴリズムの作るトポロジカル順序 σ を
$J_1, J_2, \ldots, J_n$ とする



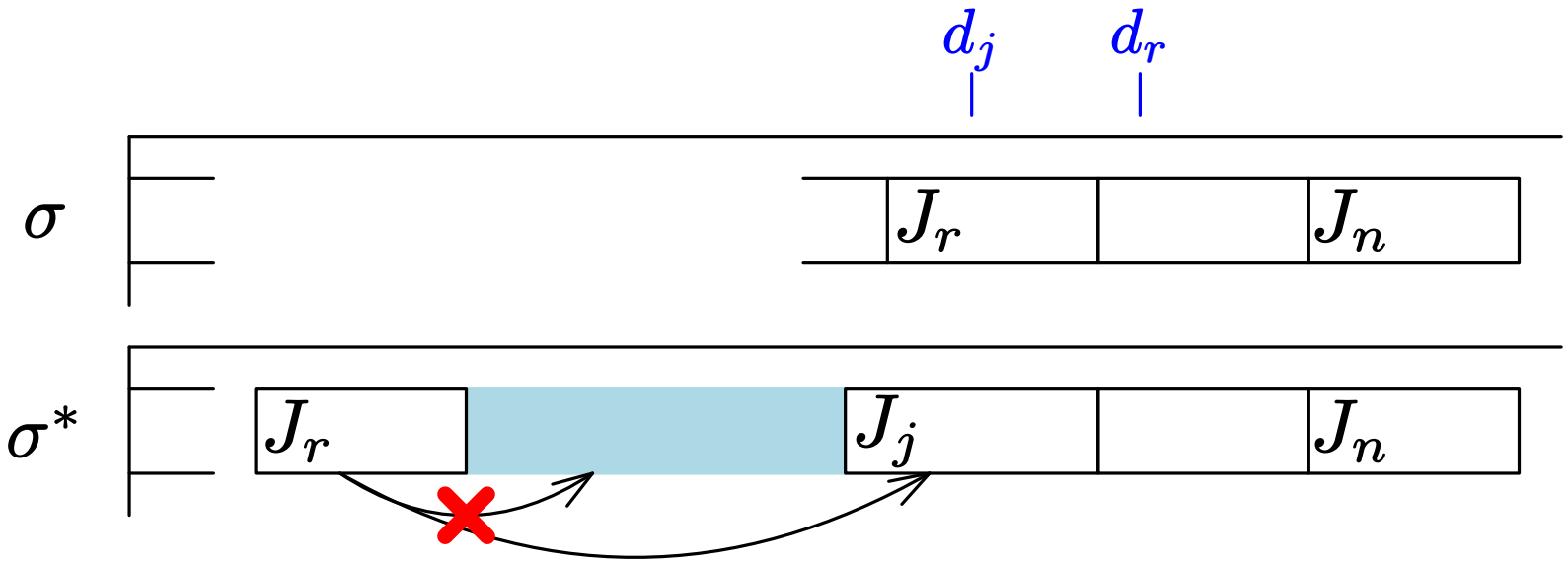
- σ^* を最適解として，完了時刻の遅い方から見て
  始めに σ と σ^* で異なるジョブに着目する
- そこにおける σ のジョブを J_r，σ^* のジョブを J_j とする
- アルゴリズムの動きから，$d_j \leq d_r$

1 | prec | $L_{\max}$: 証明 (2)



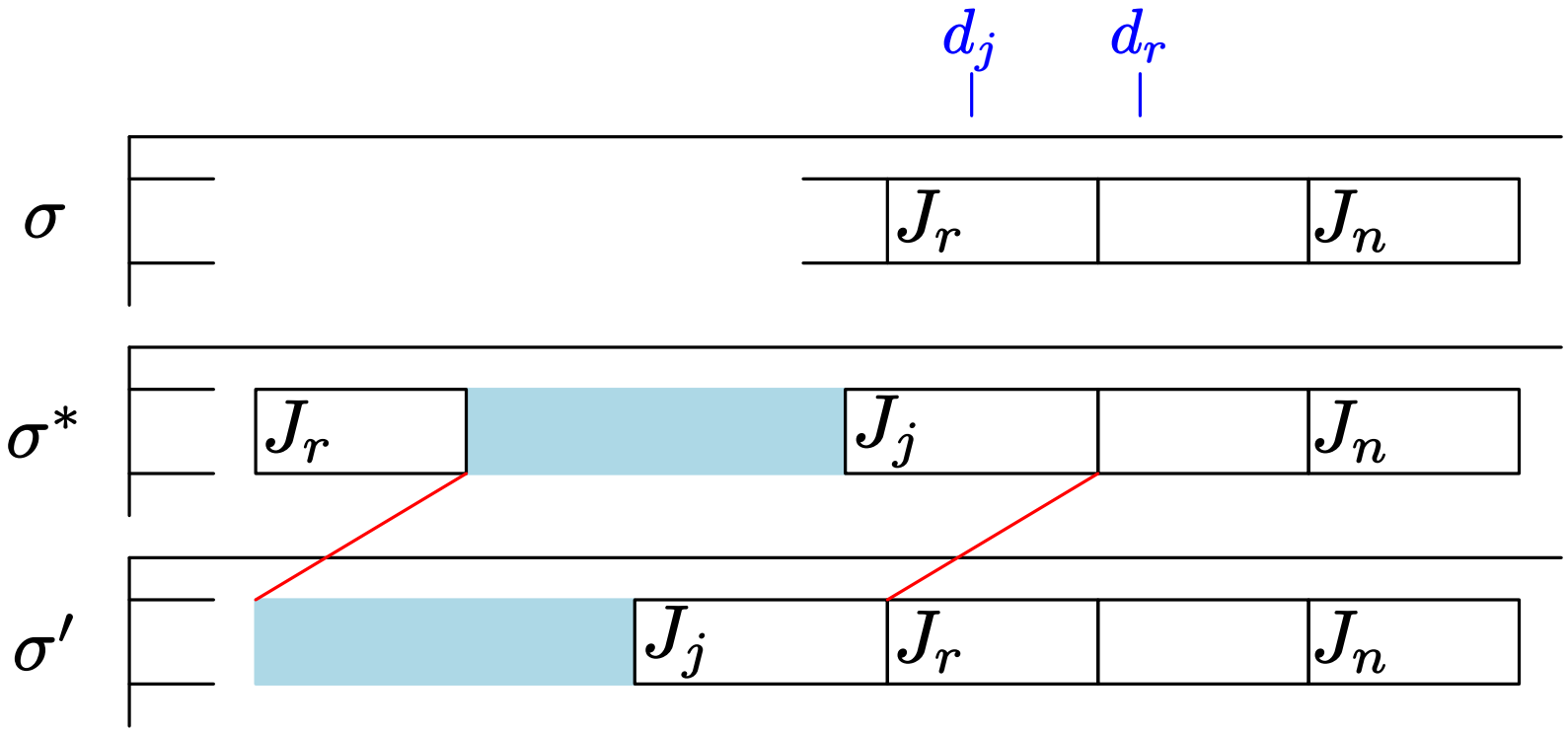
- σ^* における J_r と J_j の間のジョブと J_j を前にずらし，J_r を σ と同じ位置に置いたスケジュールを σ' とする
- $L_{\max}(\sigma') \leq L_{\max}(\sigma^*)$ なので (なぜ？)，σ' も最適解

1 | prec | $L_{\max}$: 証明 (2)



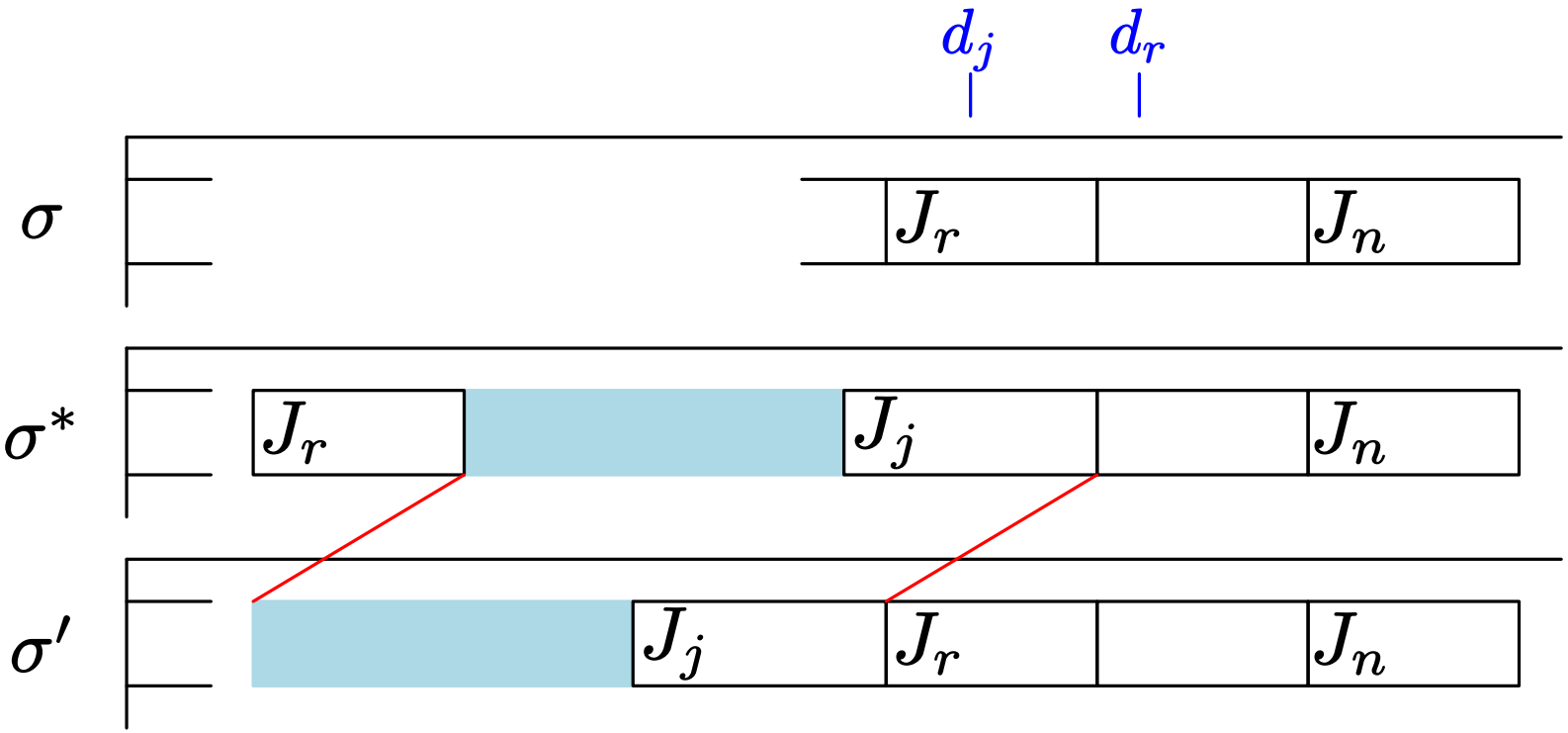
- σ^* における J_r と J_j の間のジョブと J_j を前にずらし，J_r を σ と同じ位置に置いたスケジュールを σ' とする
- $L_{\max}(\sigma') \leq L_{\max}(\sigma^*)$ なので (なぜ？)，σ' も最適解

1 | prec | $L_{\max}$: 証明 (2)



- σ^* における J_r と J_j の間のジョブと J_j を前にずらし，J_r を σ と同じ位置に置いたスケジュールを σ' とする
- $L_{\max}(\sigma') \leq L_{\max}(\sigma^*)$ なので (なぜ？)，σ' も最適解

1 | prec | $L_{\max}$: 証明 (2)



- σ^* における J_r と J_j の間のジョブと J_j を前にずらし，J_r を σ と同じ位置に置いたスケジュールを σ' とする
- $L_{\max}(\sigma') \leq L_{\max}(\sigma^*)$ なので (なぜ？)，σ' も最適解
- これを繰り返すと，σ^* から，$L_{\max}$ を増加させずに，σ を作れて，σ も最適解であると分かる □

$1 \mid \mathrm{prec} \mid \sum U_j$：強 NP 困難性

定理：$1 \mid \mathrm{prec} \mid \sum U_j$ (Garey, Johnson '76)

問題 $1 \mid \mathrm{prec} \mid \sum U_j$ は強 NP 困難

注：問題 $1 \mid\mid \sum U_j$ は $O(n \log n)$ 時間で解ける

(Moore-Hodgson のアルゴリズム)

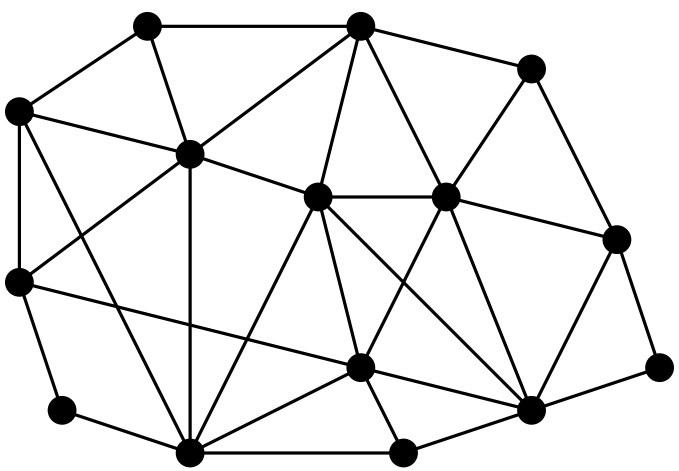
証明では **クリーク問題** を帰着する

クリーク問題

定義：クリーク問題

無向グラフ $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ と非負整数 k に対して，
$\mathcal{G}$ が頂点数 k のクリークを含むか？

クリーク (clique) = 互いに辺で結ばれた頂点の集合



観察：

クリークの頂点数 $= k \Rightarrow$

その中の辺の数 $= \binom{k}{2}$

事実 (Karp '72)

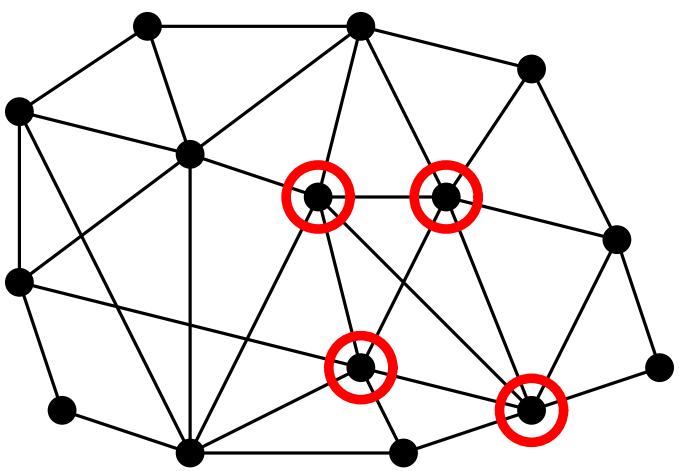
クリーク問題は (強) NP 困難

クリーク問題

定義：クリーク問題

無向グラフ $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ と非負整数 k に対して，
$\mathcal{G}$ が頂点数 k のクリークを含むか？

クリーク (clique) = 互いに辺で結ばれた頂点の集合



観察：

クリークの頂点数 $= k \Rightarrow$

その中の辺の数 $= \binom{k}{2}$

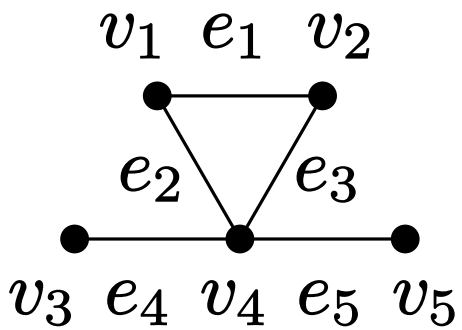
事実 (Karp '72)

クリーク問題は (強) NP 困難

$1 \mid \mathrm{prec} \mid \sum U_j$：帰着 (1)

証明：クリーク問題を帰着する

入力の変換：



$$\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$$

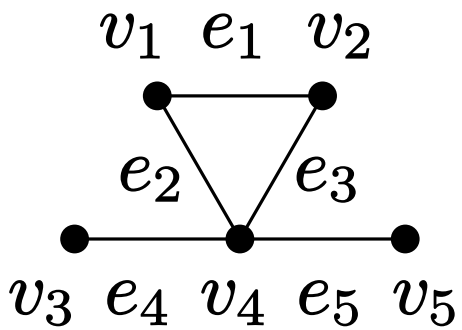
$$\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5\}$$

クリーク問題を解くアルゴリズム

$1 \mid \mathrm{prec} \mid \sum U_j$：帰着 (1)

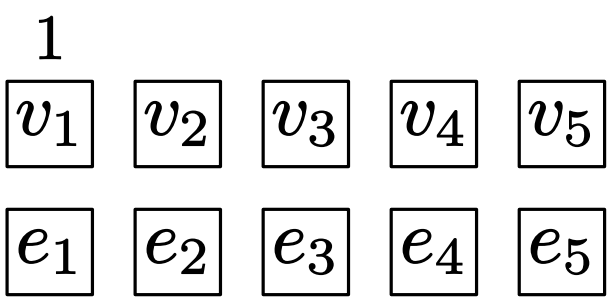
証明：クリーク問題を帰着する

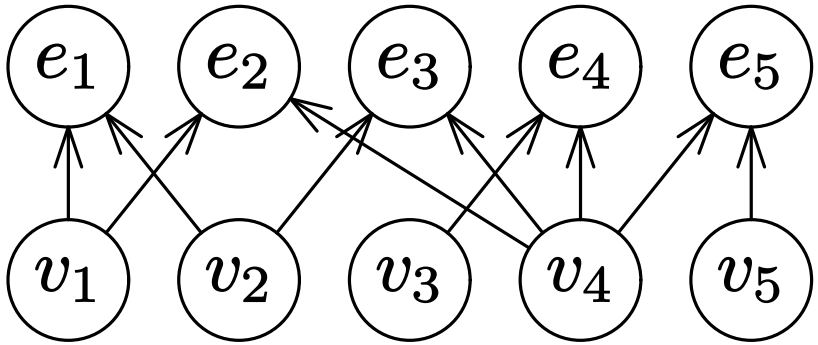
入力の変換：



$$\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$$

$$\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5\}$$



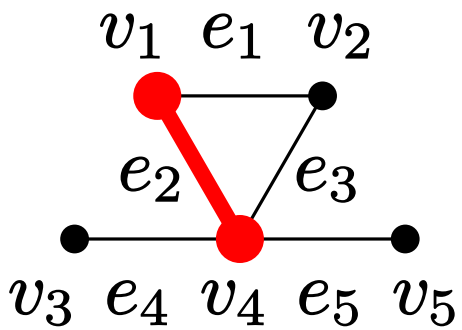


クリーク問題を解くアルゴリズム

$1 \mid \mathrm{prec} \mid \sum U_j$：帰着 (1)

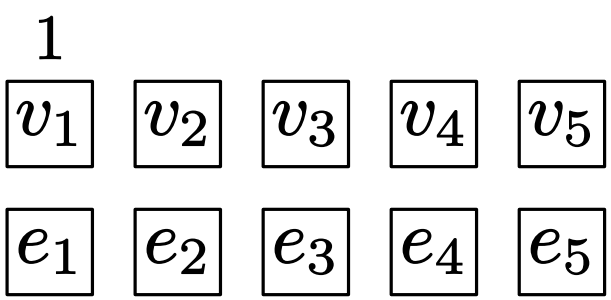
証明：クリーク問題を帰着する

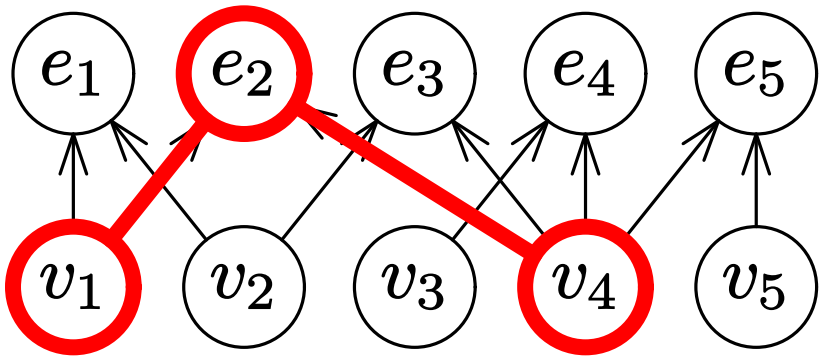
入力の変換：



$$\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$$

$$\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5\}$$



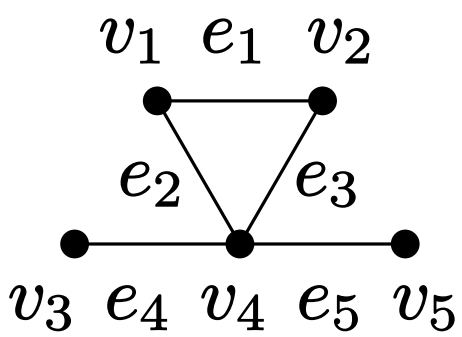


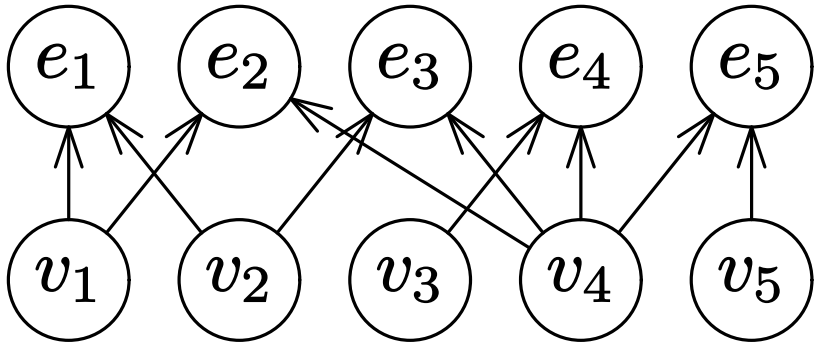
クリーク問題を解くアルゴリズム

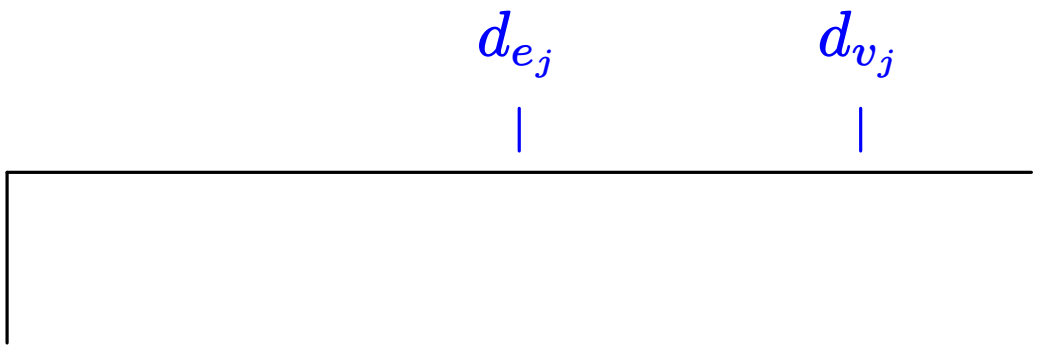
1 | prec | $\sum U_j$: 帰着 (2)

証明：クリーク問題を帰着する

入力の変換：







$$d_{e_j} = k + \binom{k}{2}$$

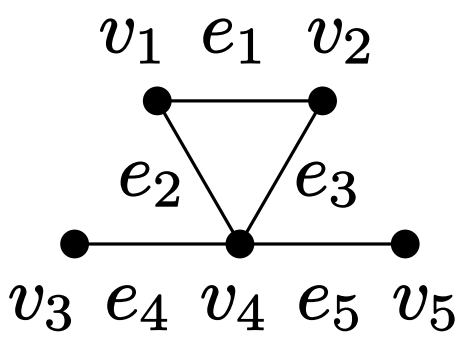
$$d_{v_j} = |\mathcal{V}| + |\mathcal{E}|$$

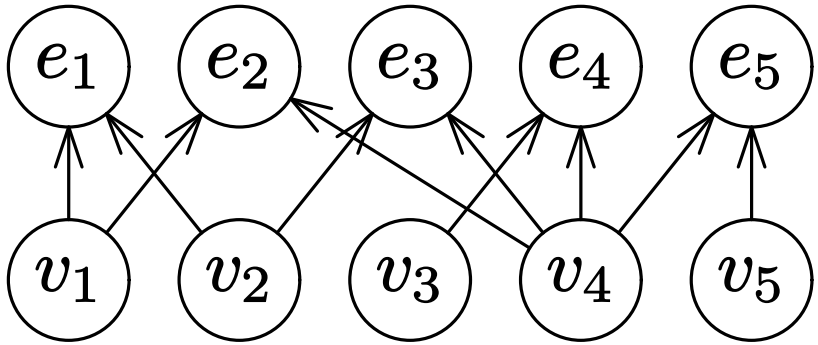
クリーク問題を解くアルゴリズム

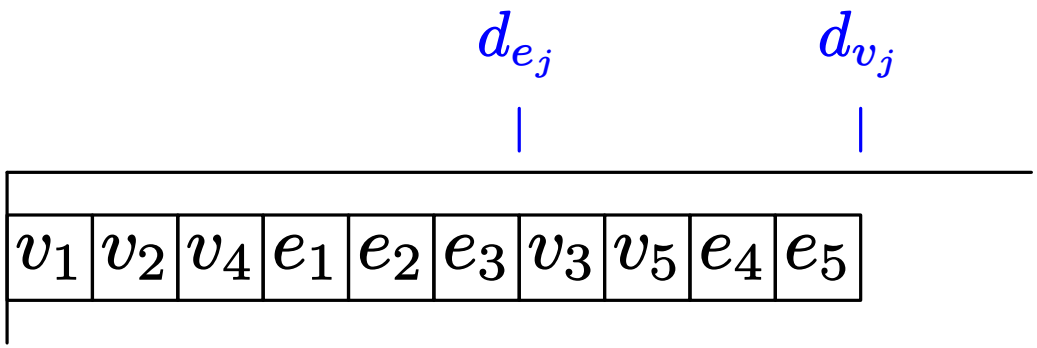
1 | prec | $\sum U_j$: 帰着 (2)

証明：クリーク問題を帰着する

入力の変換：







$$d_{e_j} = k + \binom{k}{2}$$

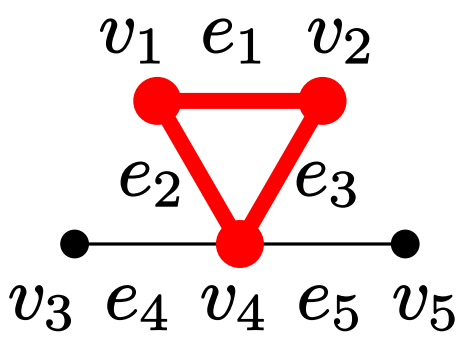
$$d_{v_j} = |\mathcal{V}| + |\mathcal{E}|$$

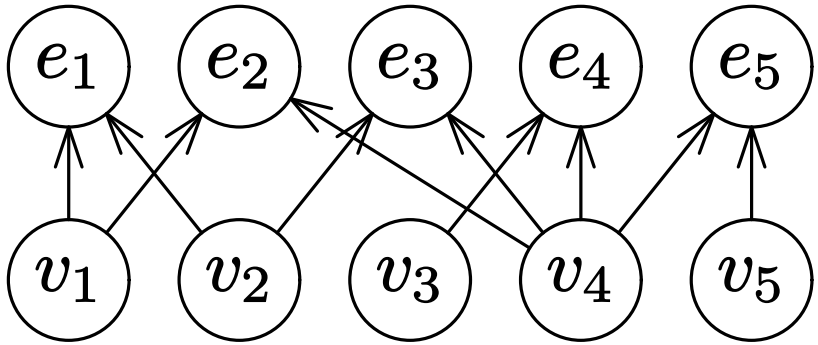
クリーク問題を解くアルゴリズム

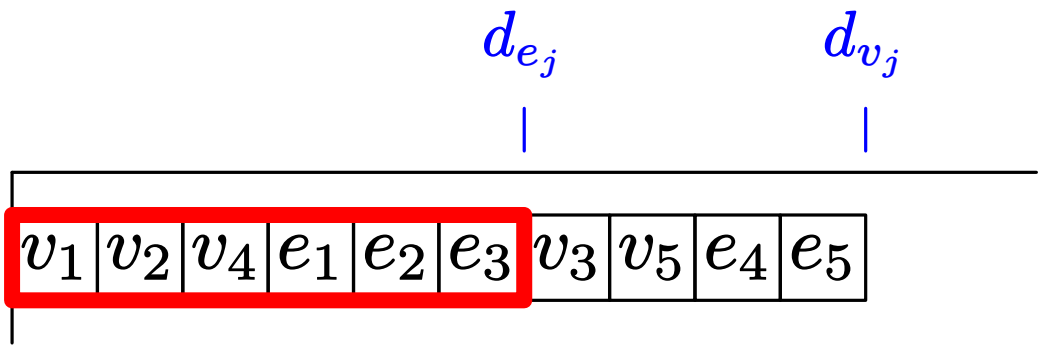
1 | prec | $\sum U_j$: 帰着 (2)

証明：クリーク問題を帰着する

入力の変換：







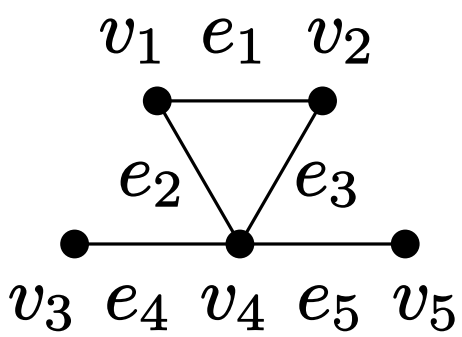
$$d_{e_j} = k + \binom{k}{2}$$

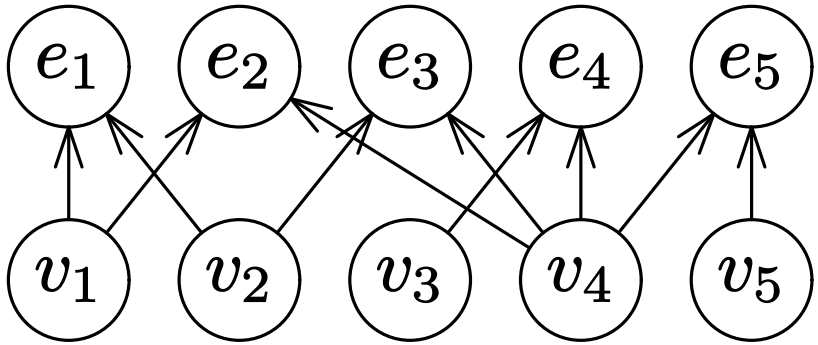
$$d_{v_j} = |\mathcal{V}| + |\mathcal{E}|$$

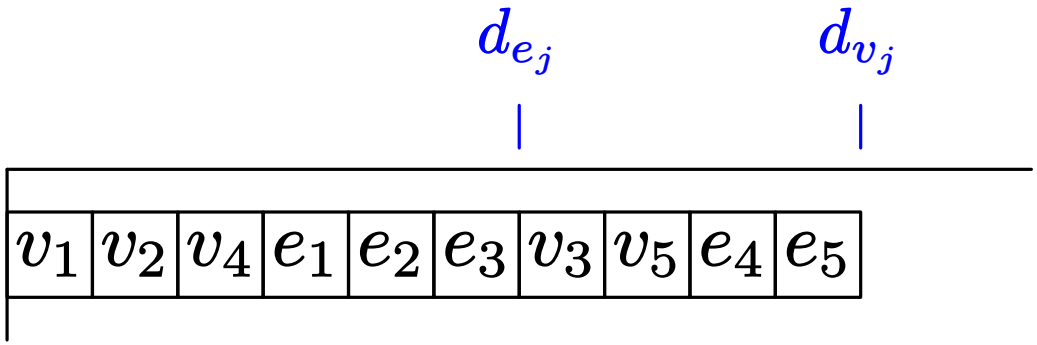
クリーク問題を解くアルゴリズム

証明：クリーク問題を帰着する

入力の変換：







$$d_{e_j} = k + \binom{k}{2}$$

$$d_{v_j} = |\mathcal{V}| + |\mathcal{E}|$$

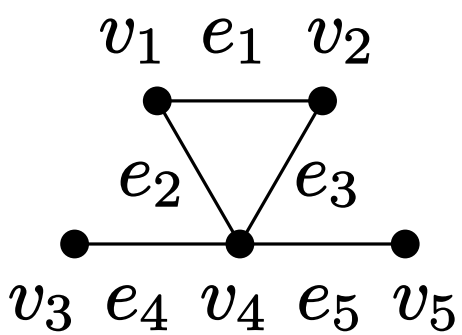
出力の変換：

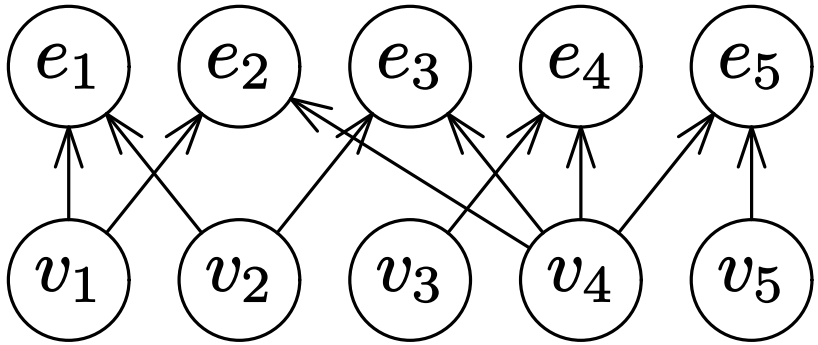
最適値 $\leq |\mathcal{E}| - \binom{k}{2} \Rightarrow$ 頂点数 k のクリークがある

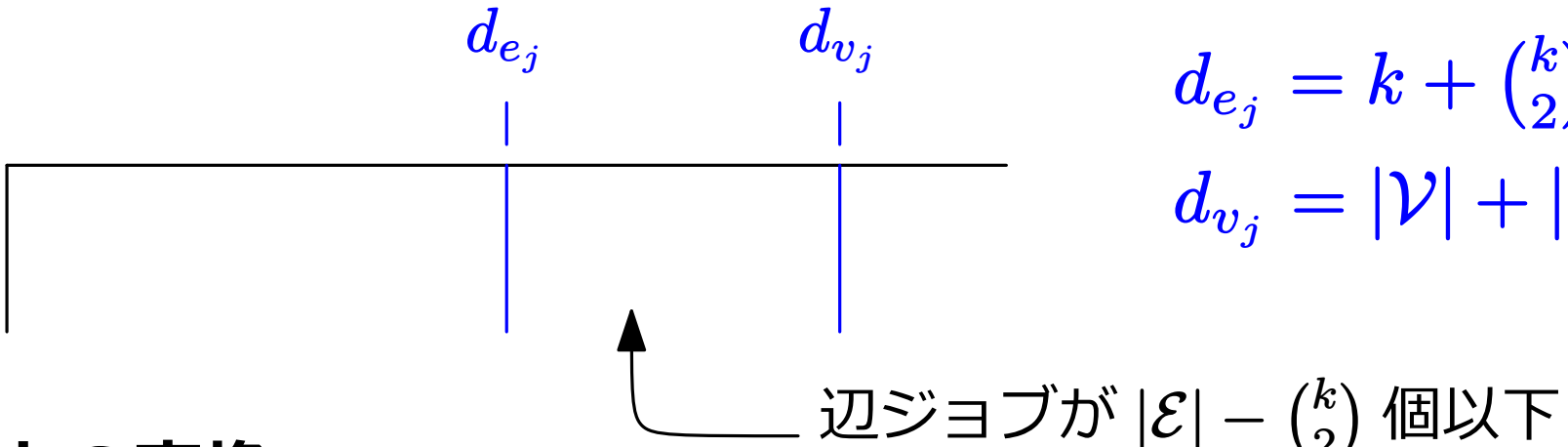
最適値 $> |\mathcal{E}| - \binom{k}{2} \Rightarrow$ 頂点数 k のクリークがない

証明：クリーク問題を帰着する

入力の変換：







$$d_{e_j} = k + \binom{k}{2}$$

$$d_{v_j} = |\mathcal{V}| + |\mathcal{E}|$$

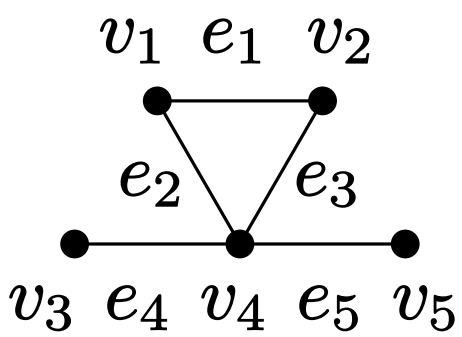
出力の変換：

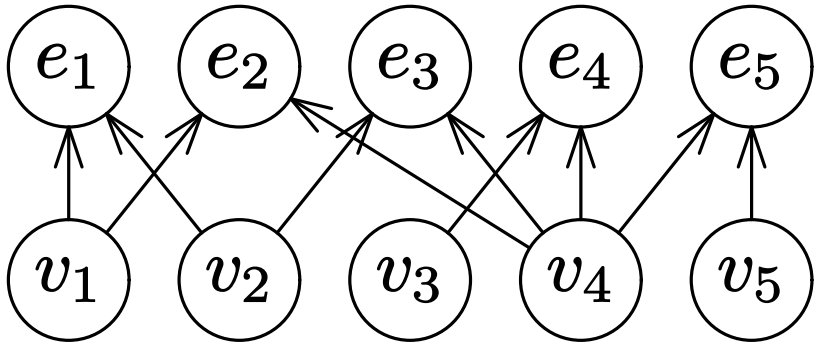
最適値 $\leq |\mathcal{E}| - \binom{k}{2} \Rightarrow$ 頂点数 k のクリークがある

最適値 $> |\mathcal{E}| - \binom{k}{2} \Rightarrow$ 頂点数 k のクリークがない

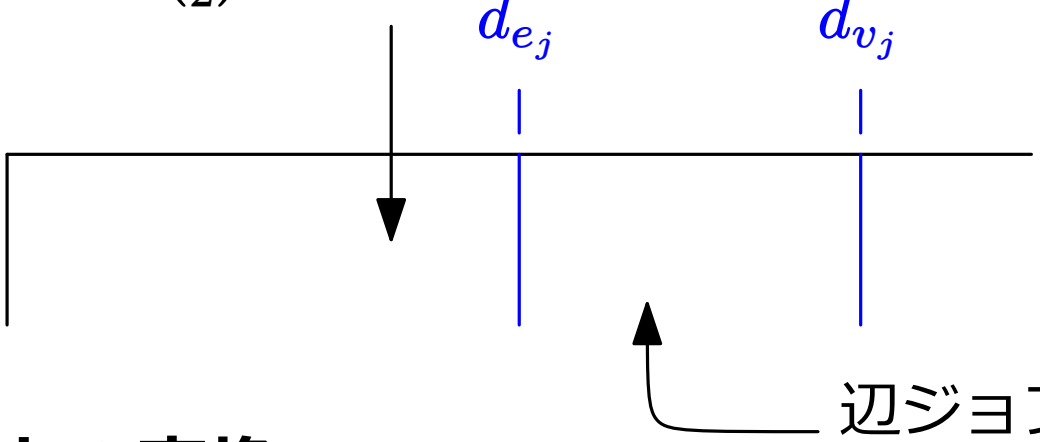
証明：クリーク問題を帰着する

入力の変換：





辺ジョブが $\binom{k}{2}$ 個以上



$$d_{e_j} = k + \binom{k}{2}$$

$$d_{v_j} = |\mathcal{V}| + |\mathcal{E}|$$

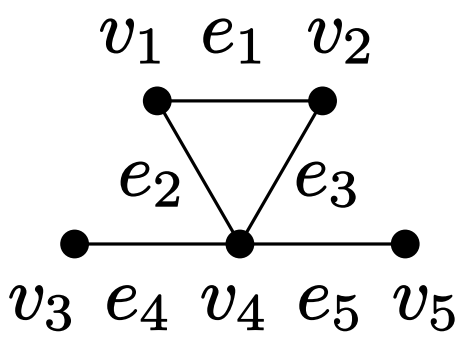
出力の変換：

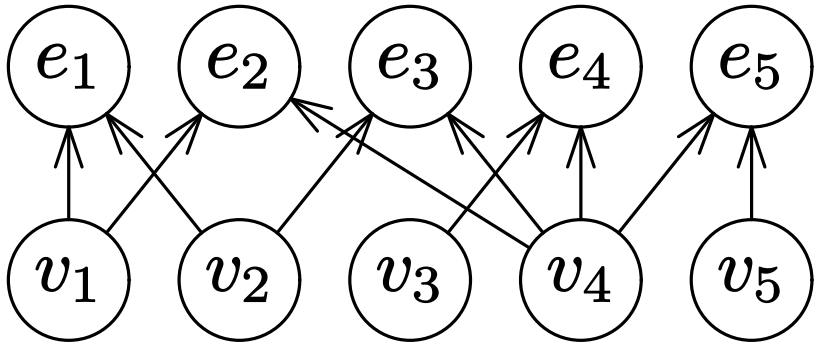
最適値 $\leq |\mathcal{E}| - \binom{k}{2} \Rightarrow$ 頂点数 k のクリークがある

最適値 $> |\mathcal{E}| - \binom{k}{2} \Rightarrow$ 頂点数 k のクリークがない

証明：クリーク問題を帰着する

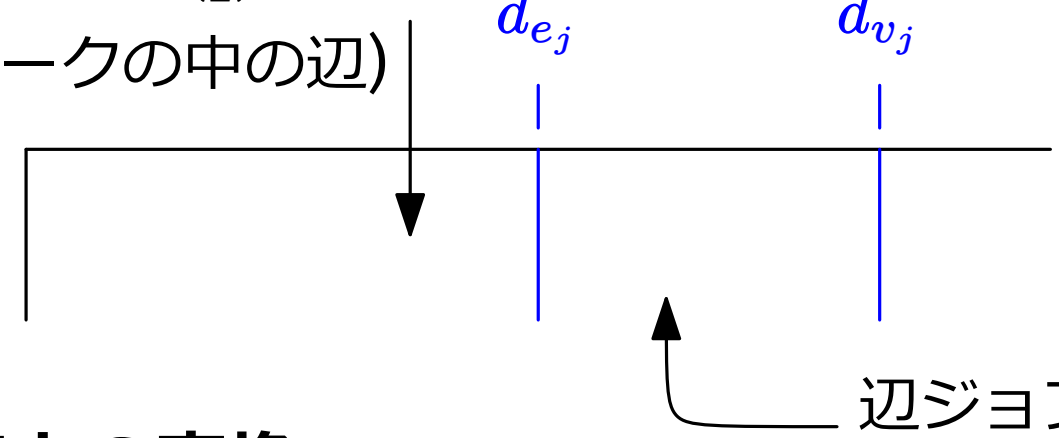
入力の変換：





辺ジョブが $\binom{k}{2}$ 個以上

(クリークの中の辺)



$$d_{e_j} = k + \binom{k}{2}$$

$$d_{v_j} = |\mathcal{V}| + |\mathcal{E}|$$

出力の変換：

最適値 $\leq |\mathcal{E}| - \binom{k}{2} \Rightarrow$ 頂点数 k のクリークがある

最適値 $> |\mathcal{E}| - \binom{k}{2} \Rightarrow$ 頂点数 k のクリークがない □

[再掲] まとめ：一機械，先行制約あり

まとめ : $1 \mid \mathrm{prec} \mid \gamma$

γ		計算量	(文献)
$C_{\max}$	最大完了時刻	$O(n+h)$	
$\sum C_j$	総完了時刻	強 NP 困難	(Lawler '78)
$L_{\max}$	最大納期ずれ	$O(n \log n + h)$	(Lawler '73)
$\sum T_j$	総納期遅れ	強 NP 困難	(Lenstra, Rinnooy Kan '78)
$\sum U_j$	納期遅れジョブ数	強 NP 困難	(Garey, Johnson '76)

[再掲] まとめ：一機械，先行制約あり

まとめ：$1 \mid \mathrm{prec} \mid \gamma$

γ		計算量	参考：先行制約なし
$C_{\max}$	最大完了時刻	$O(n+h)$	$O(n)$
$\sum C_j$	総完了時刻	強 NP 困難	$O(n \log n)$
$L_{\max}$	最大納期ずれ	$O(n \log n + h)$	$O(n \log n)$
$\sum T_j$	総納期遅れ	強 NP 困難	弱 NP 困難
$\sum U_j$	納期遅れジョブ数	強 NP 困難	$O(n \log n)$

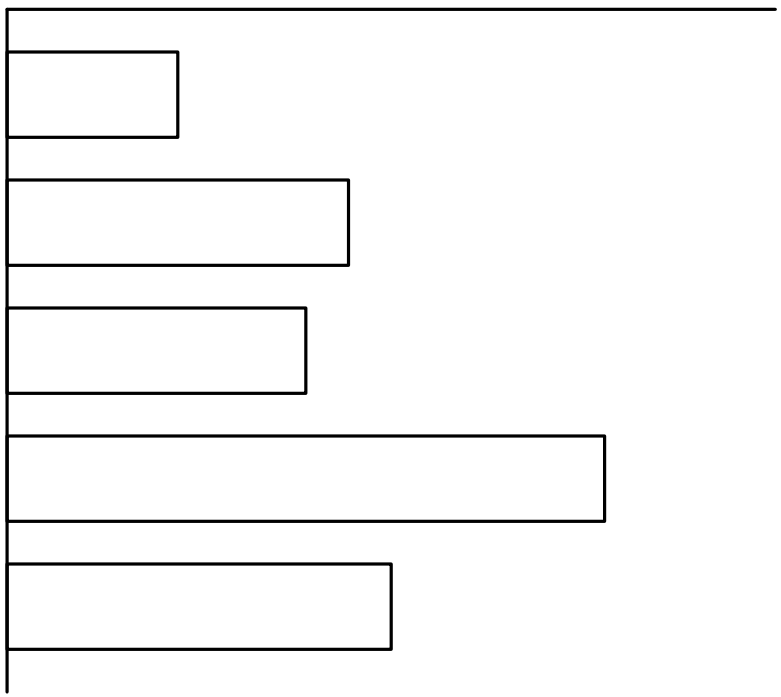
本日の内容

1. 先行制約の基礎概念
2. 一機械スケジューリングと先行制約
3. $P\infty \mid \textbf{prec} \mid C_{\max}$ **とクリティカル・パス法**

- J. E. Kelley, M. R. Walker, Critical-path planning and scheduling. *Proceedings of the Eastern Joint Computer Conference* (1959) pp. 160–173.

機械数が ∞ ？

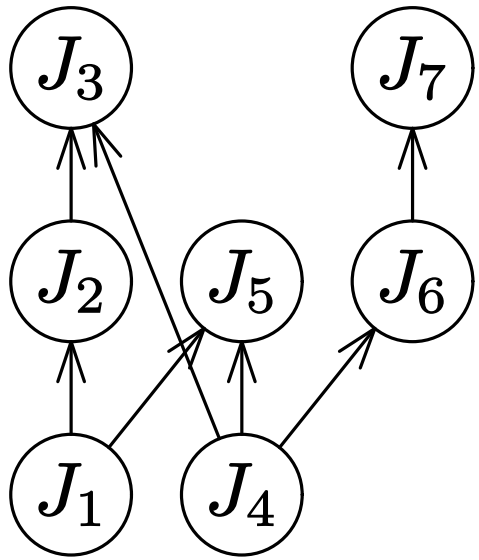
先行制約がない場合 $\rightsquigarrow$ つまらない

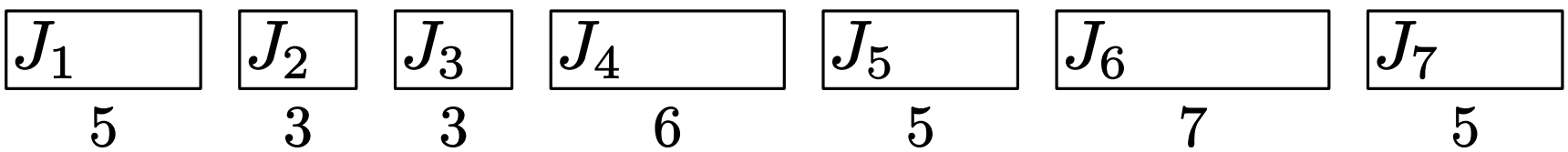


先行制約がある場合 $\rightsquigarrow$ ？ ？ ？

クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く

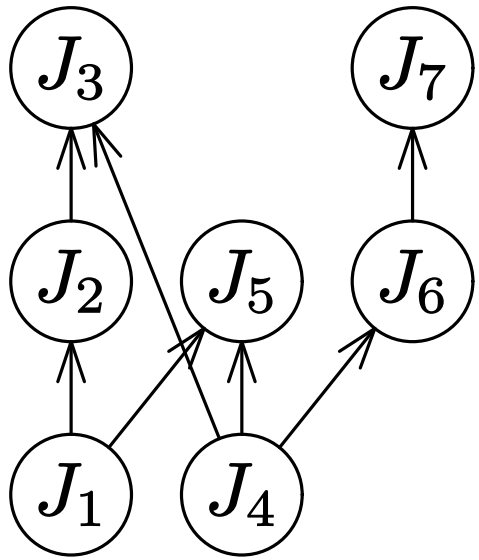


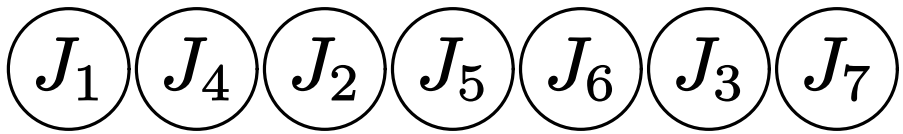


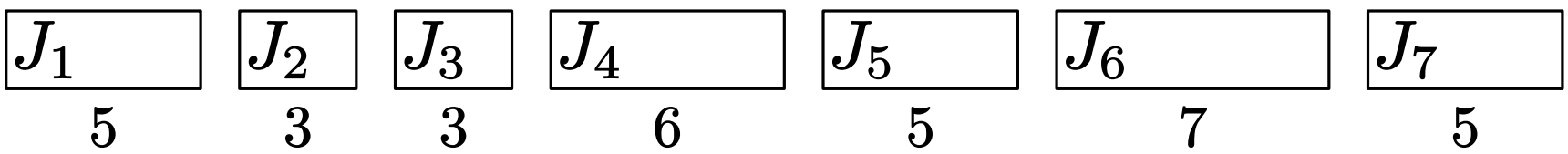
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く



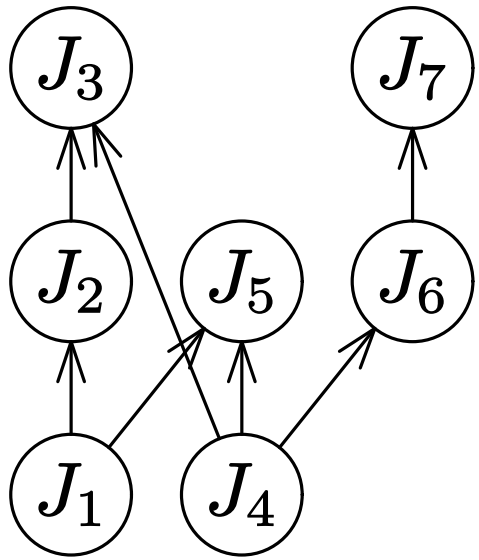


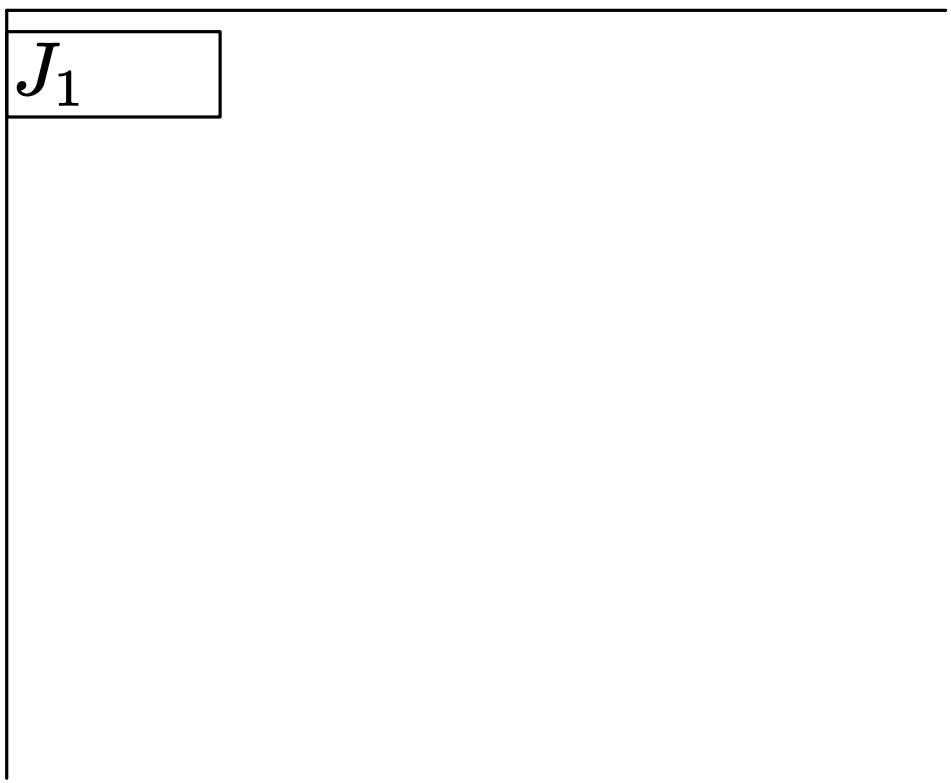


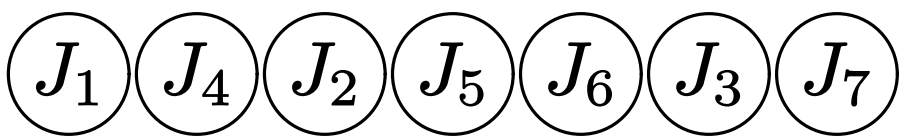
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

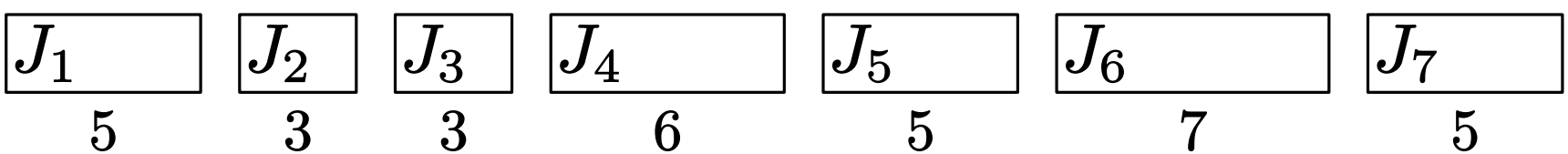
クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





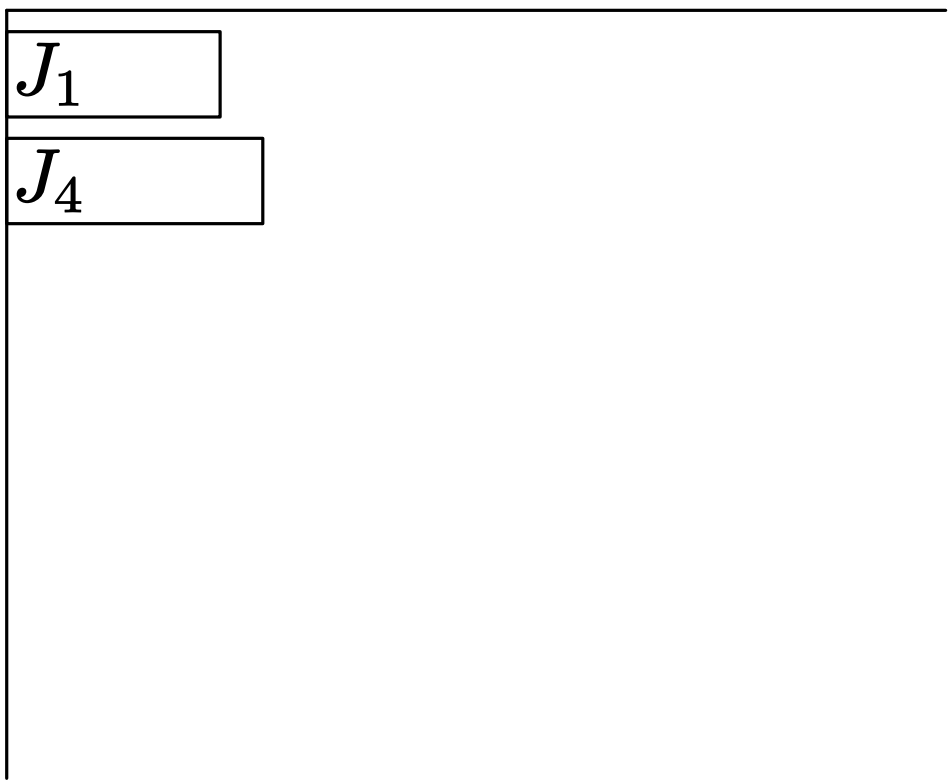


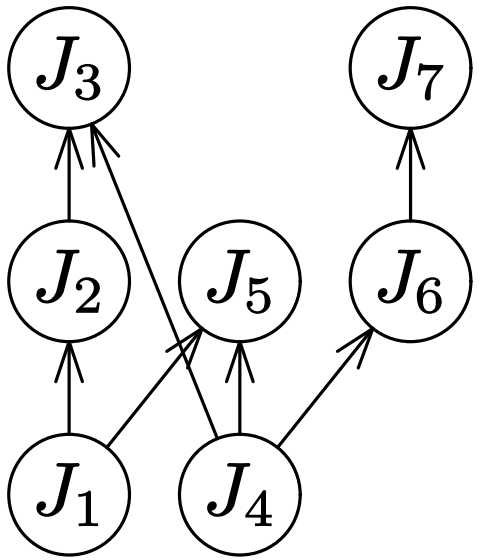


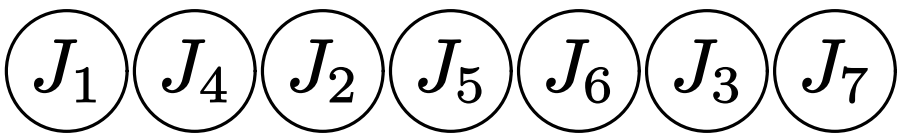
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

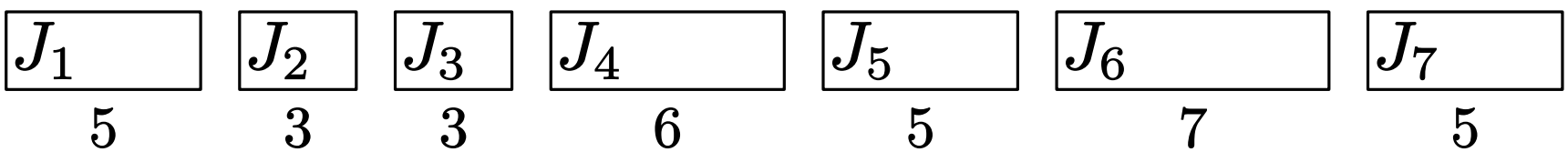
クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





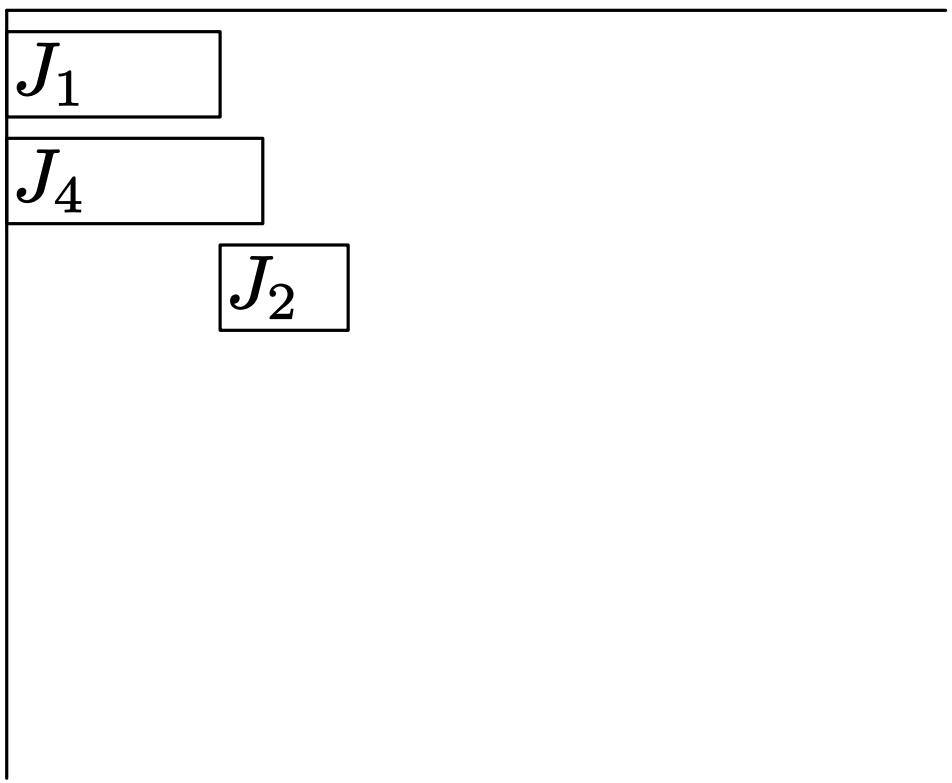


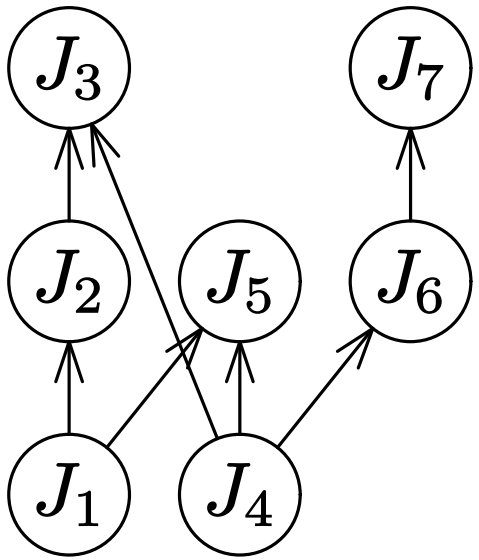


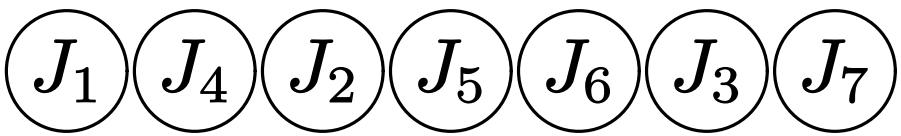
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

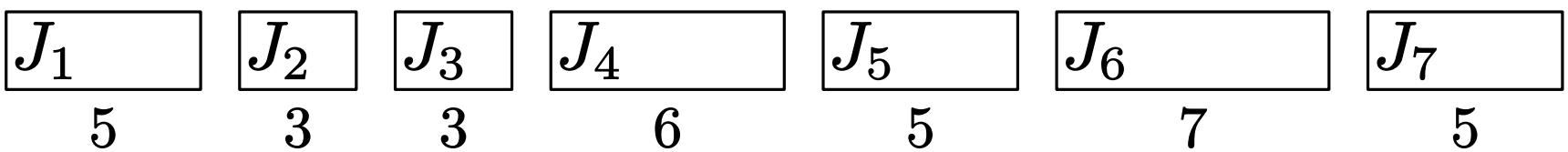
クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





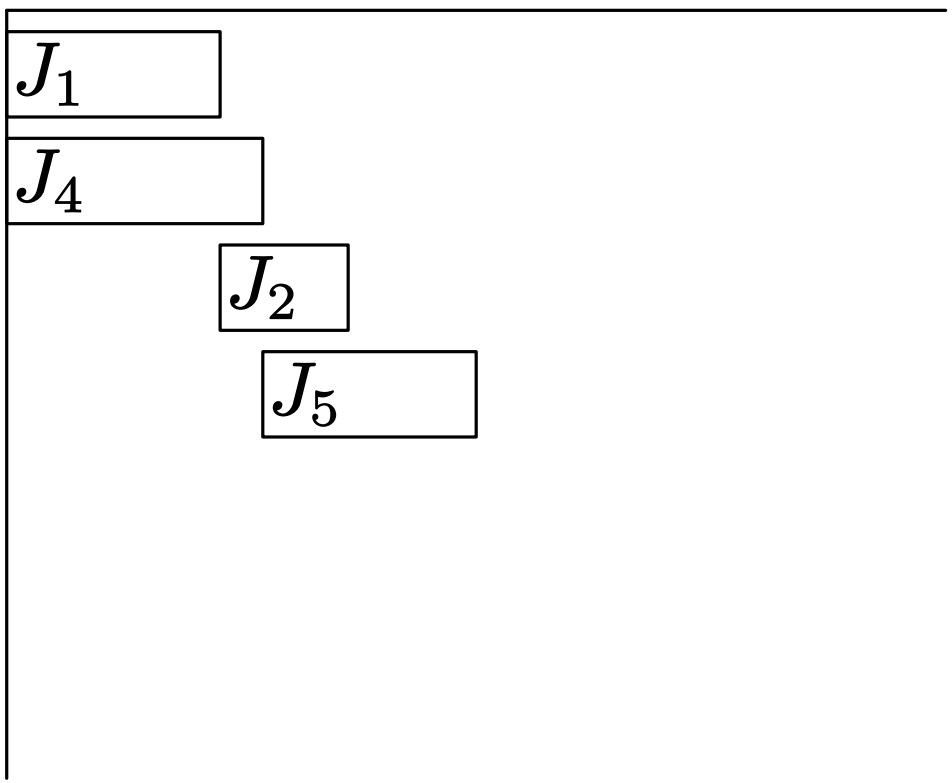


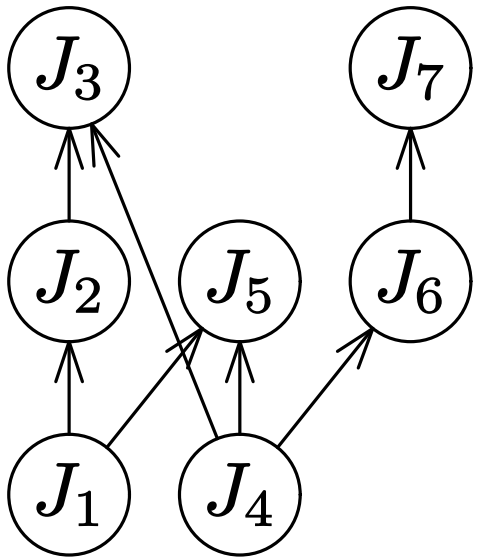


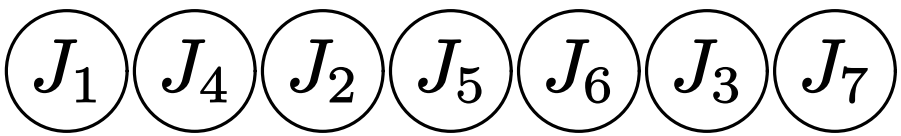
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

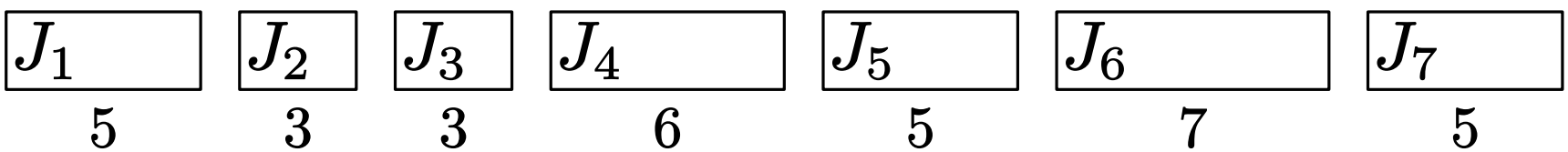
クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





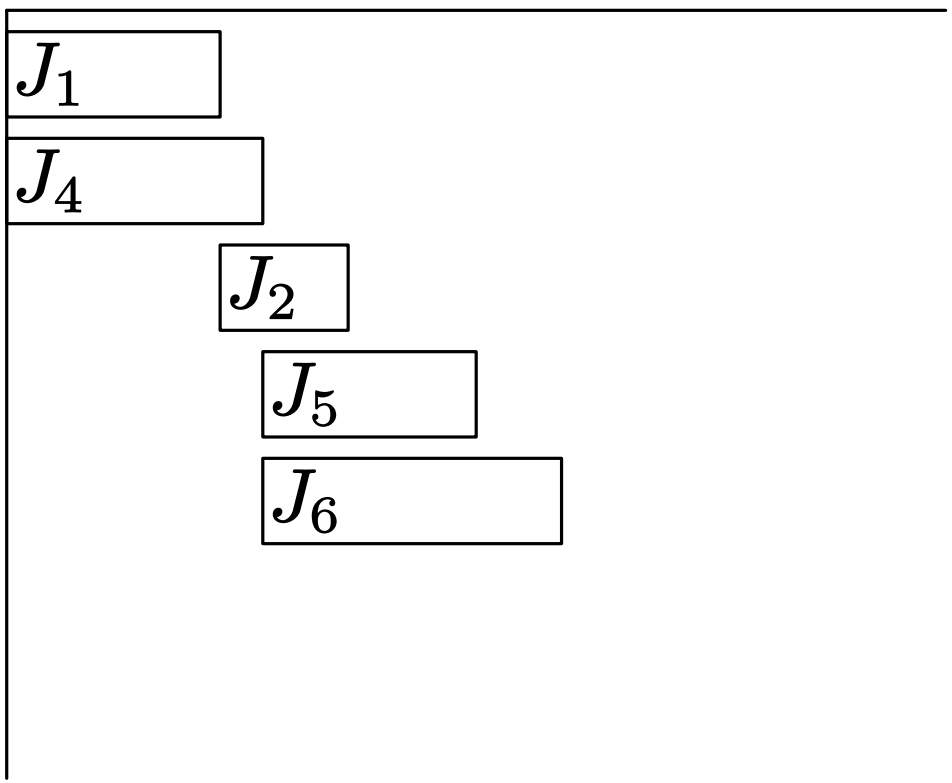


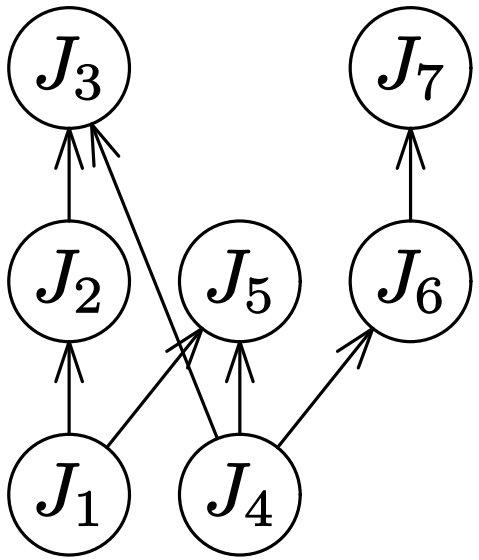


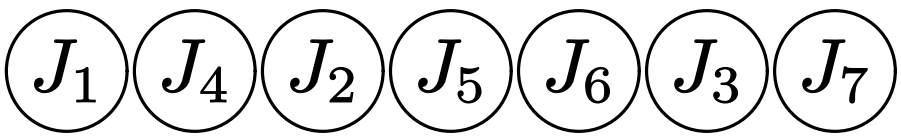
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

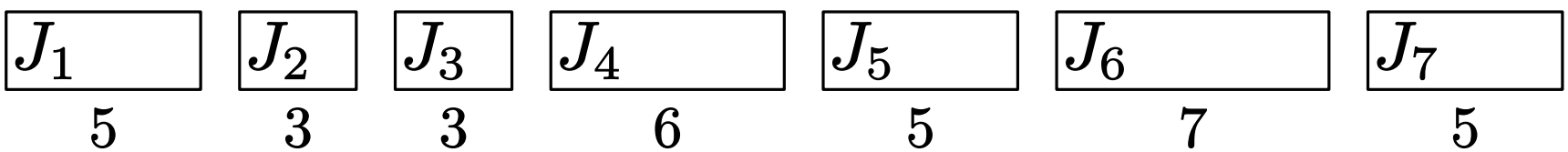
クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





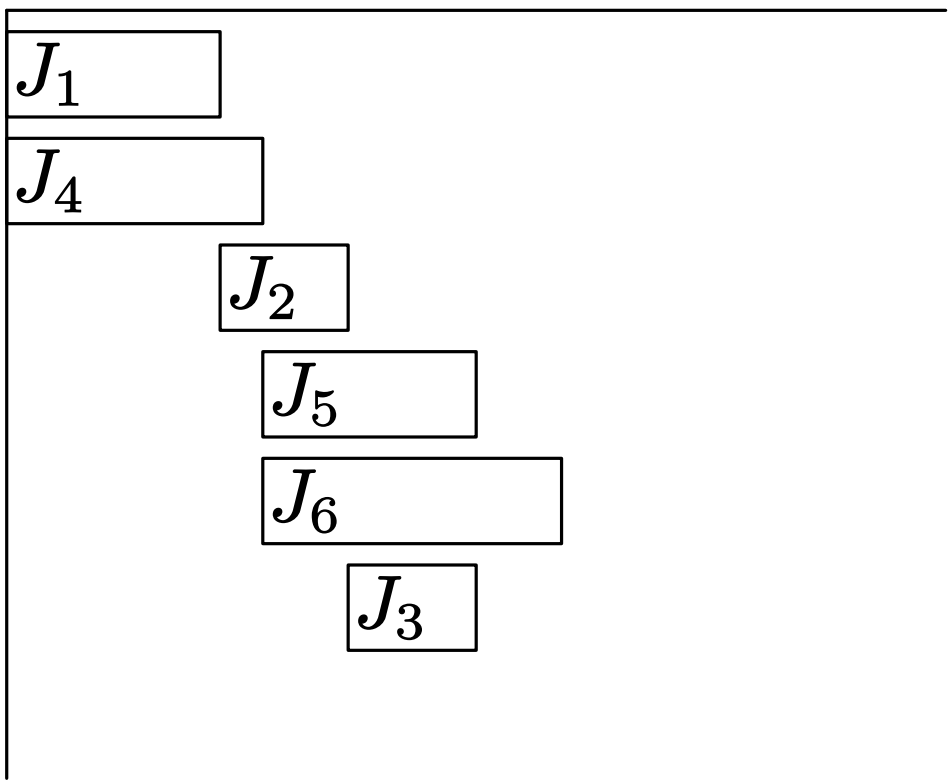


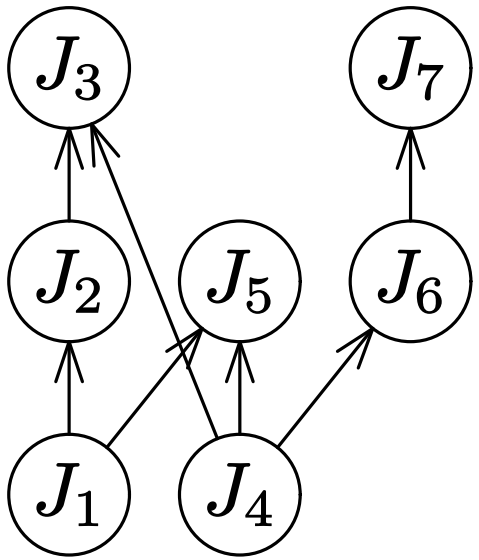


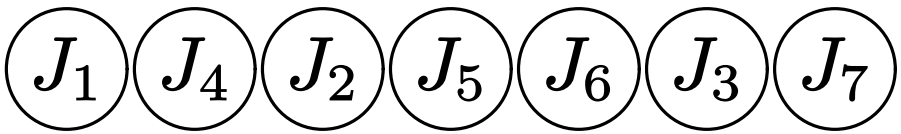
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

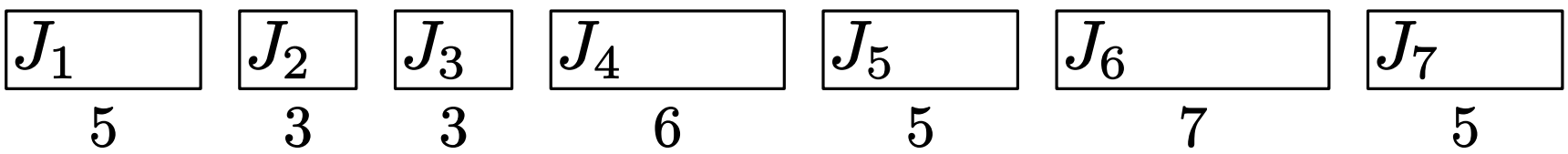
クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





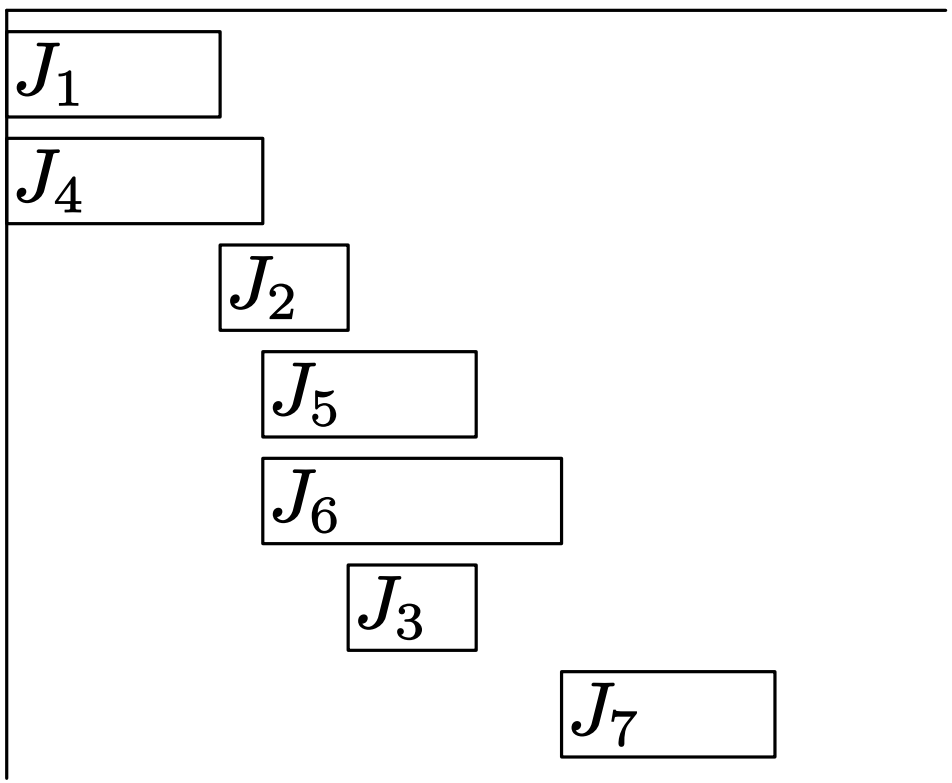


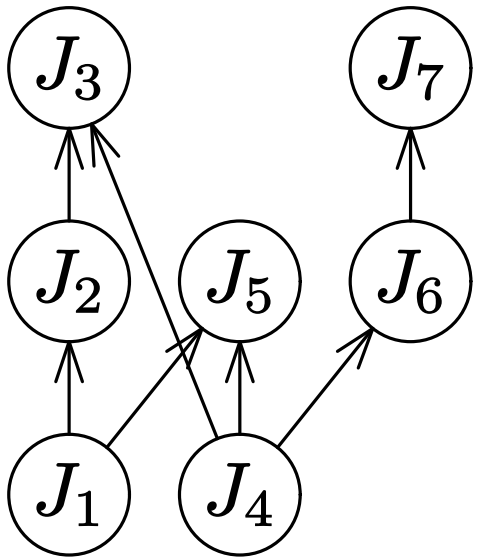


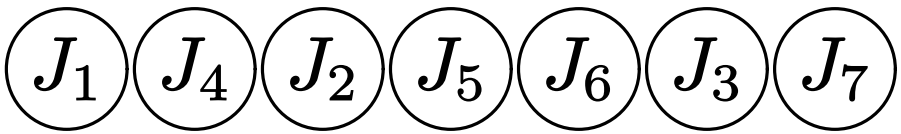
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

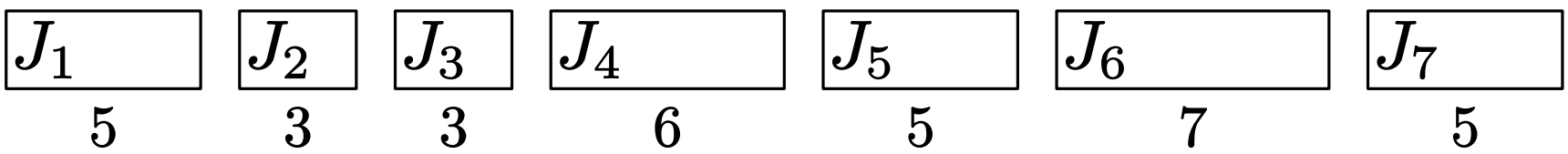
クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





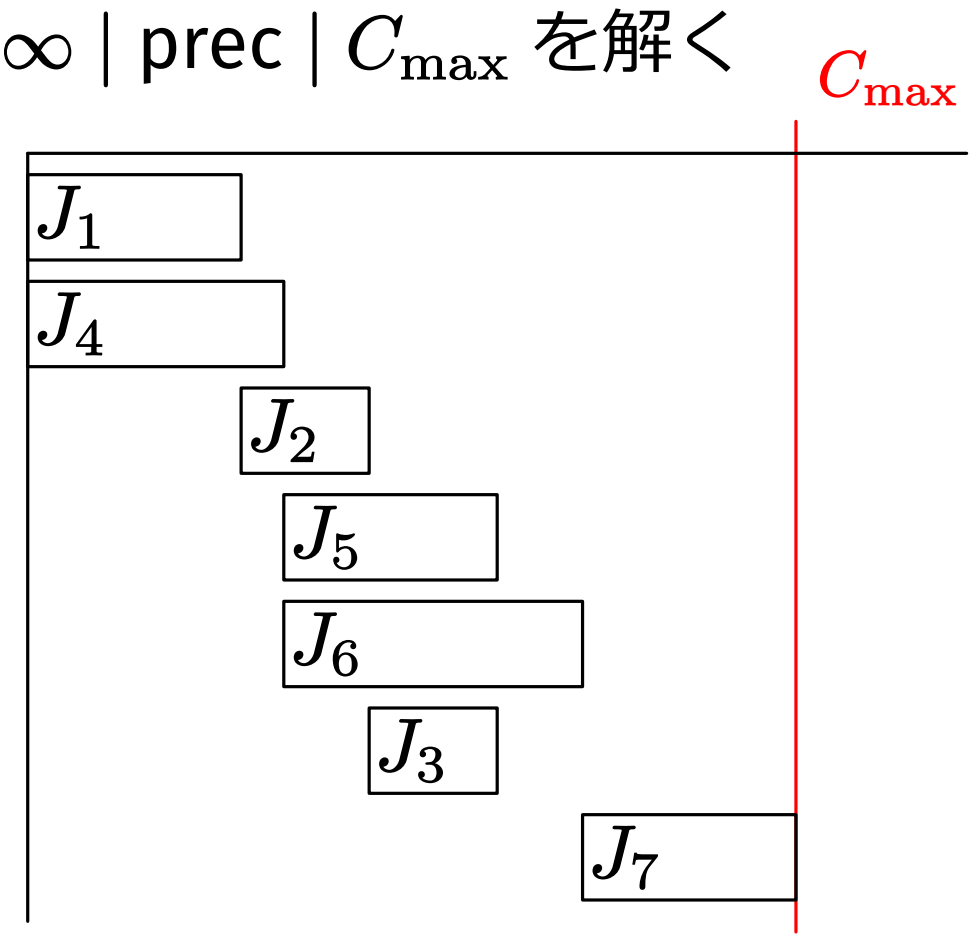


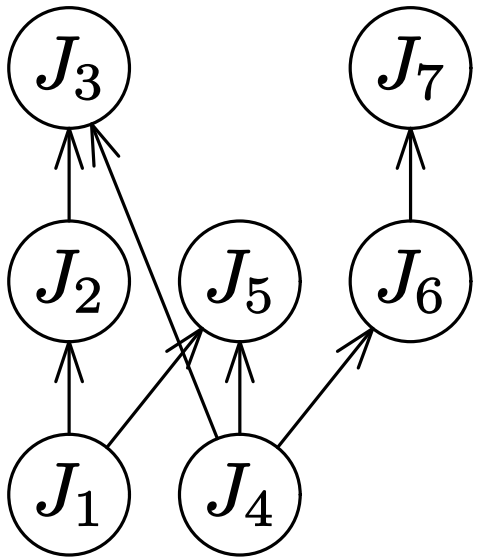


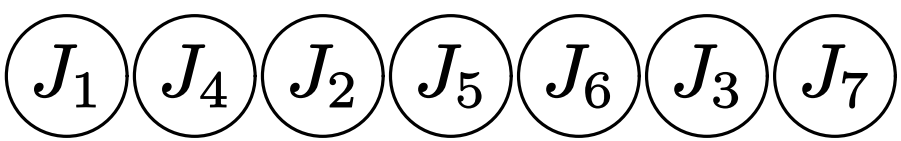
注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

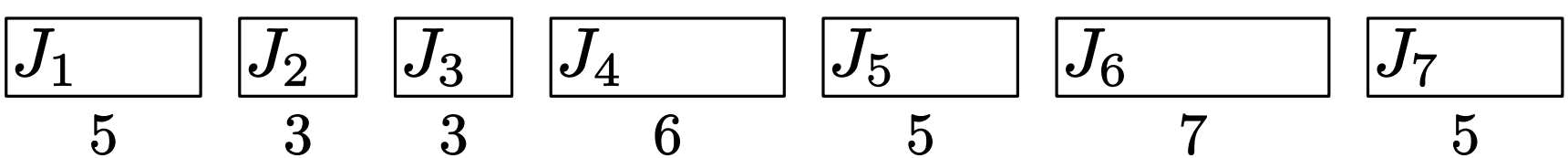
クリティカル・パス法：例

$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





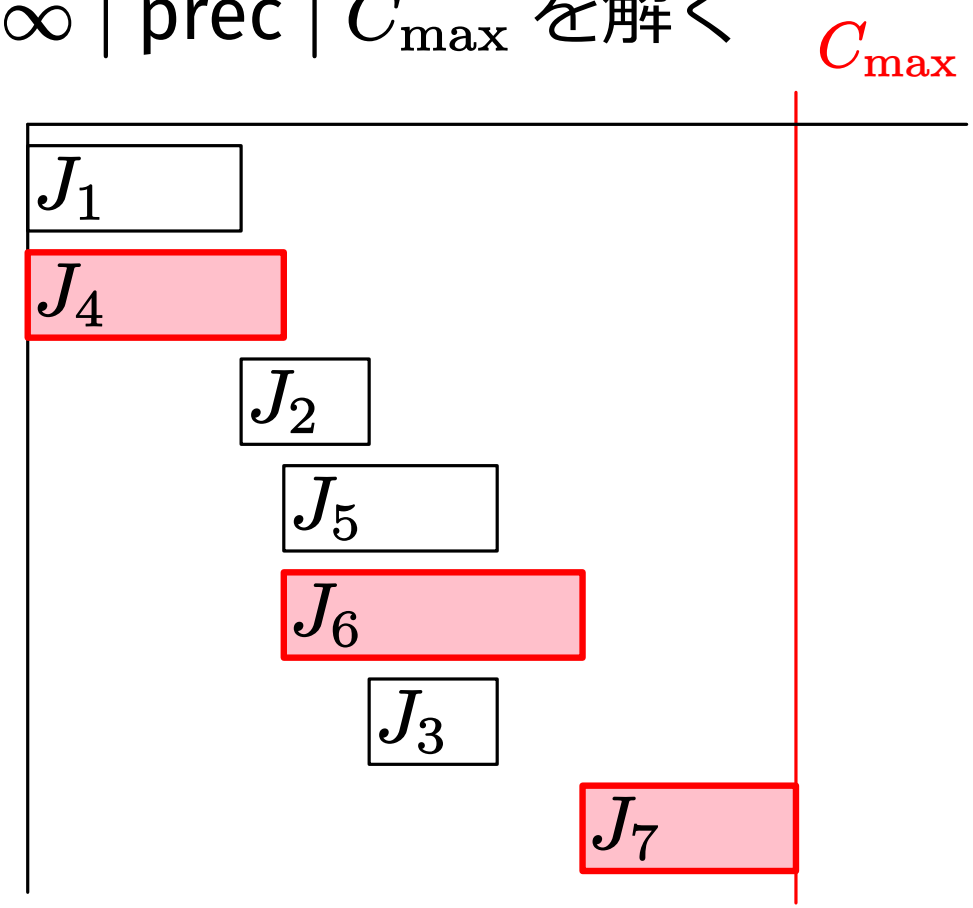


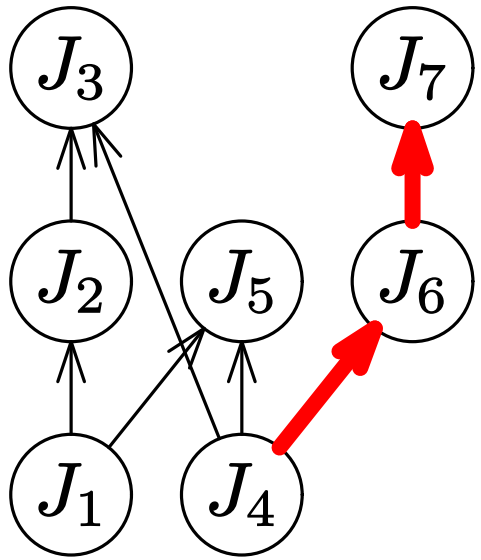


注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

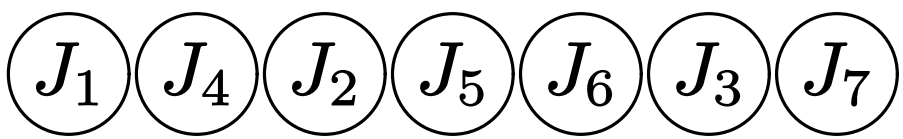
クリティカル・パス法：例

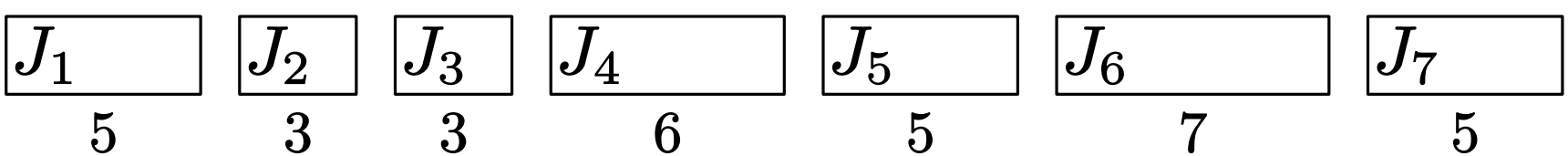
$P\infty \mid \mathrm{prec} \mid C_{\max}$ を解く





クリティカル・パス





注：クリティカル・パス法は普通「アロー・ダイアグラム」で説明されるが，ここではそうしていない

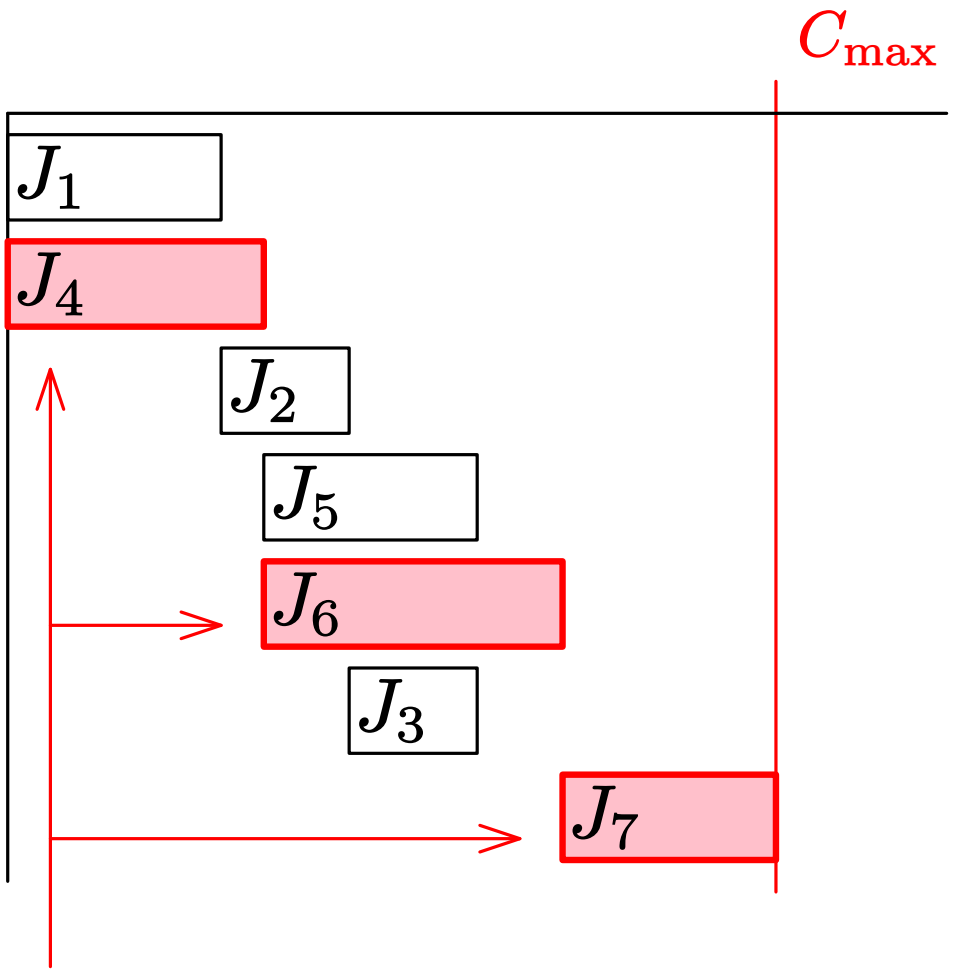
クリティカル・パス法 (critical path method) 32/44

クリティカル・パス法 (Kelley, Walker '59)

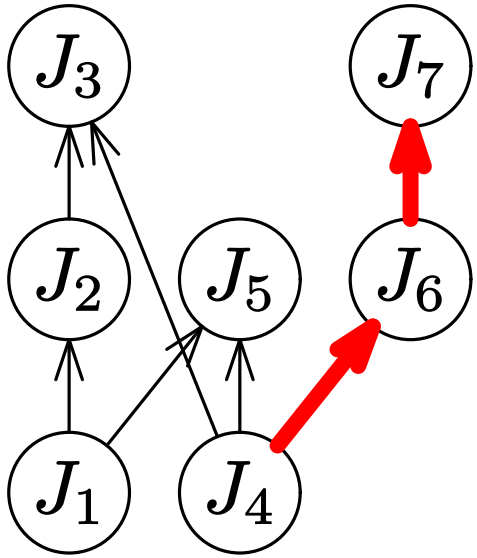
1. トポロジカル順序 σ を 1 つ求める
2. σ に沿って，ジョブ J_j に次を行う
   - $J_k \to J_j$ となる J_k の完了時刻の最大値を J_j の開始時刻とする

計算量：$O(n+h)$

クリティカル，スラック



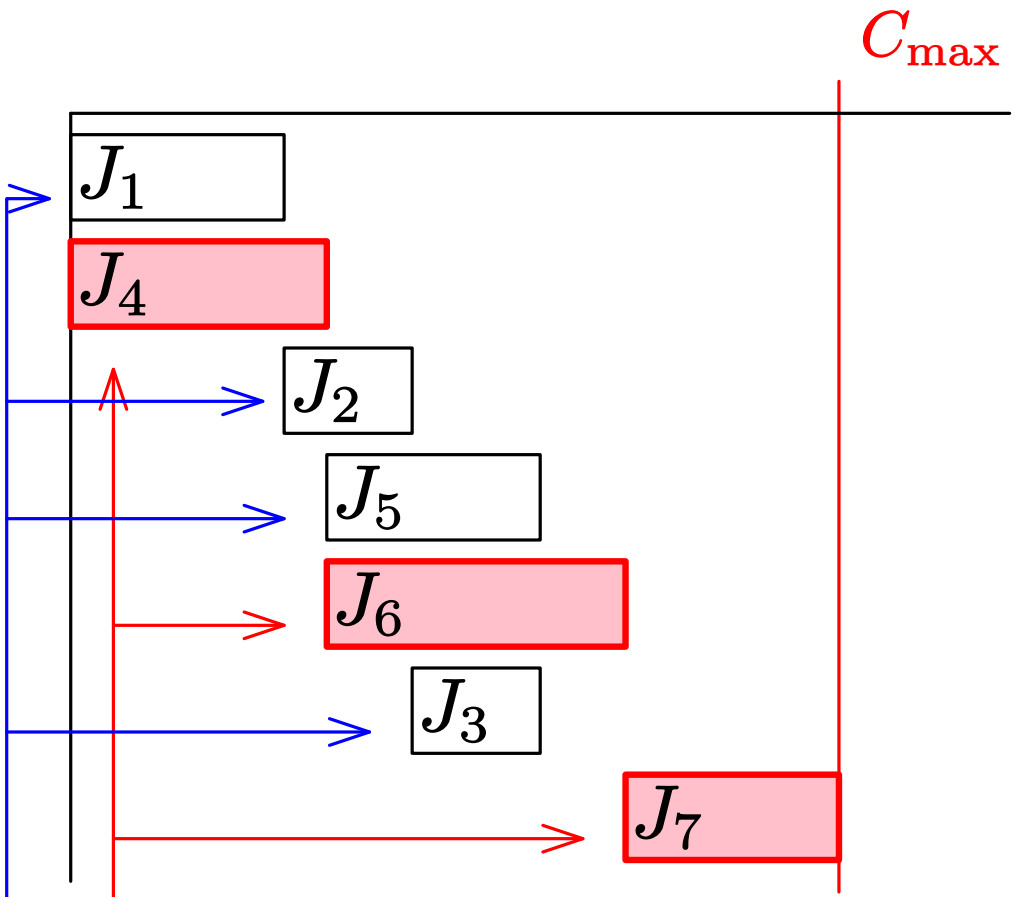
クリティカル・ジョブ (critical job)

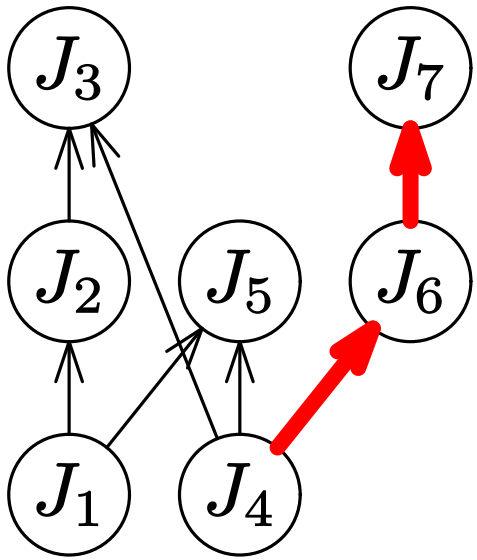


クリティカル・パス

critical = 臨界的な，致命的な

クリティカル，スラック





クリティカル・パス

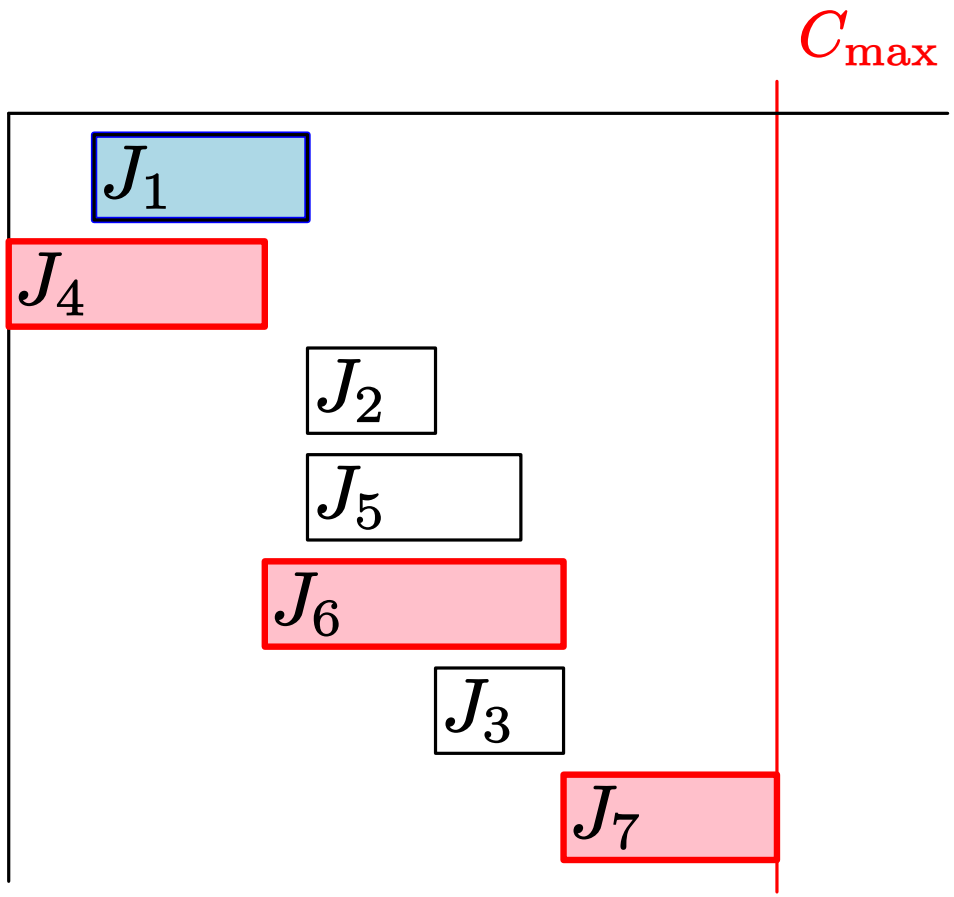
critical = 臨界的な，致命的な

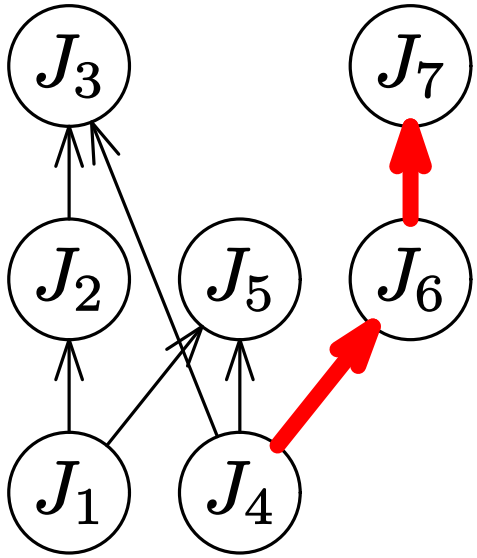
クリティカル・ジョブ (critical job)

slack = 緩い，怠慢な

スラック・ジョブ (slack job)

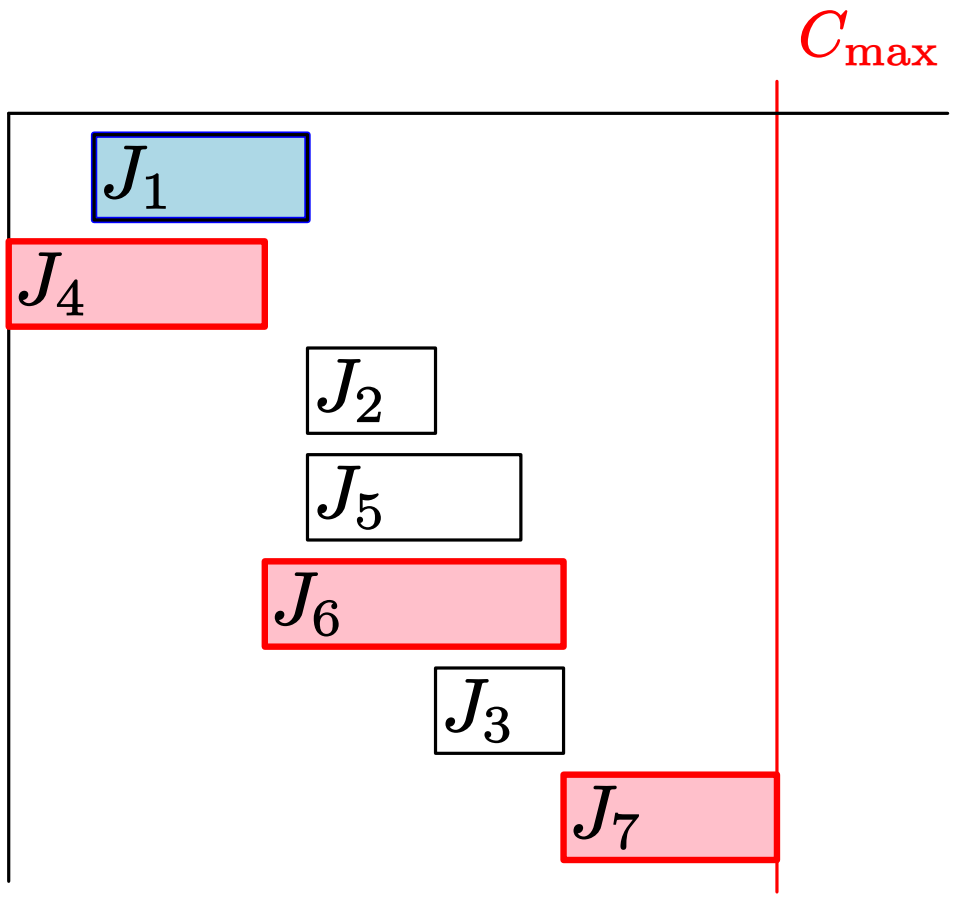
スラック・ジョブの処理が遅れたら？

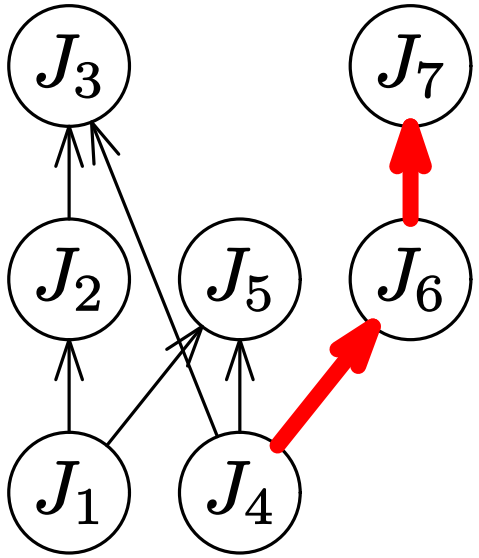




クリティカル・パス

スラック・ジョブの処理が遅れたら？



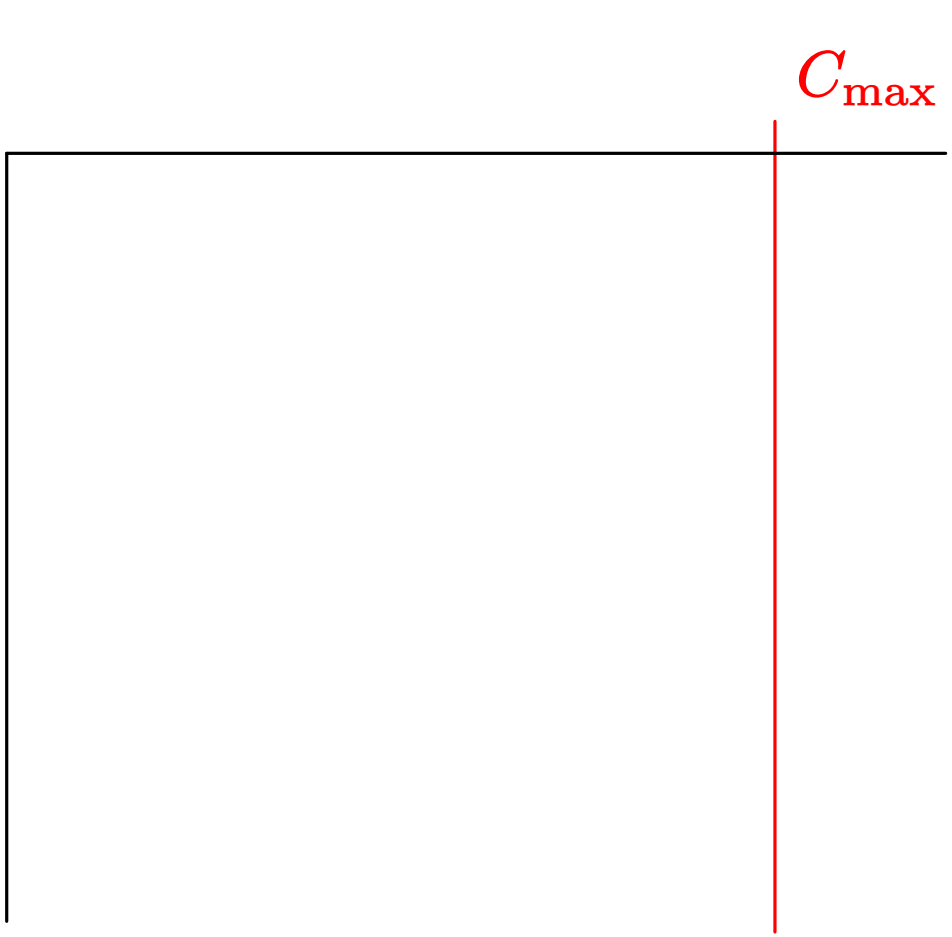


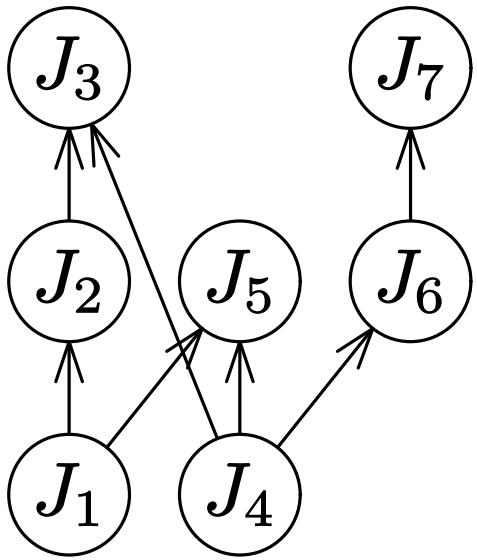
クリティカル・パス

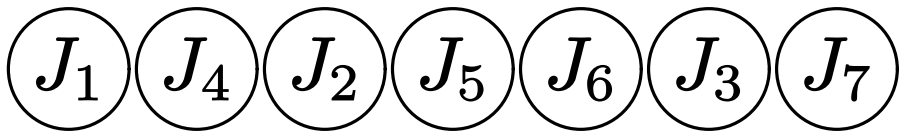
疑問

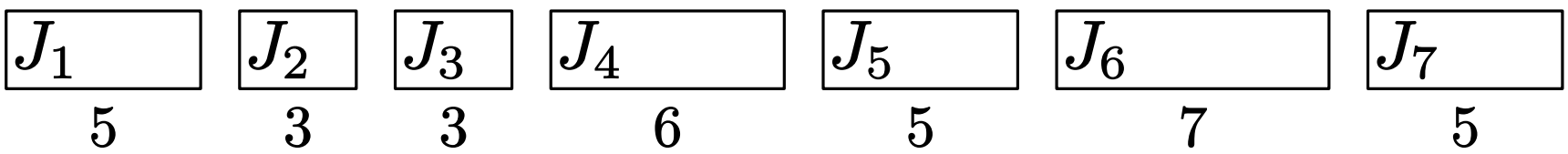
遅れても最大完了時刻が保たれるのは，どんなときか？

「後ろ」から決めていく

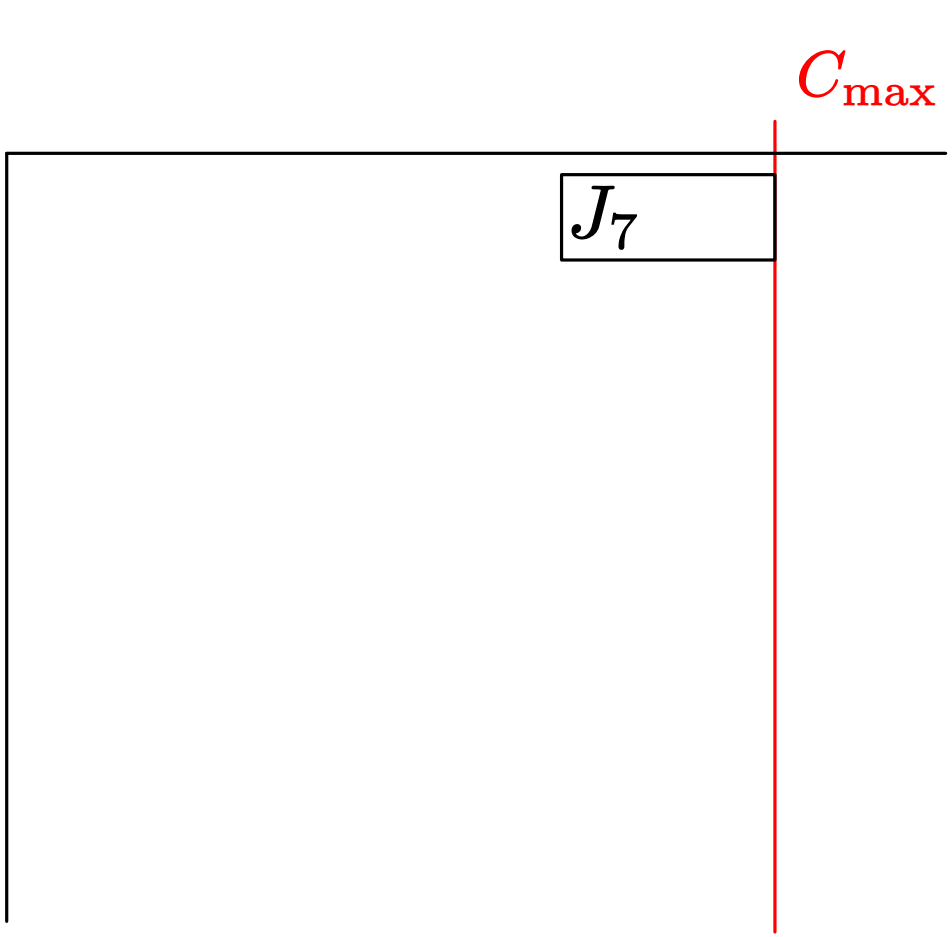

$C_{\max}$

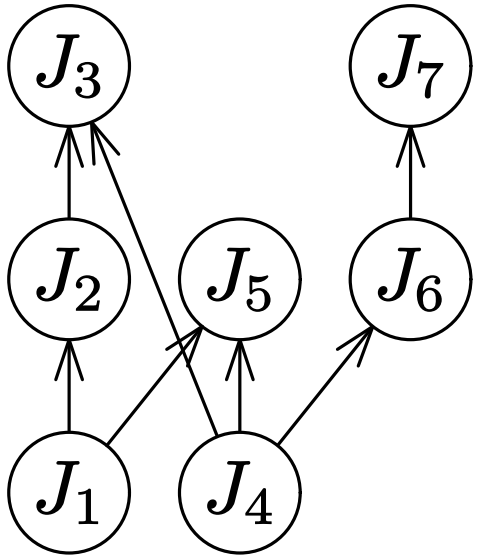

J_3
J_7
J_2
J_5
J_6
J_1
J_4

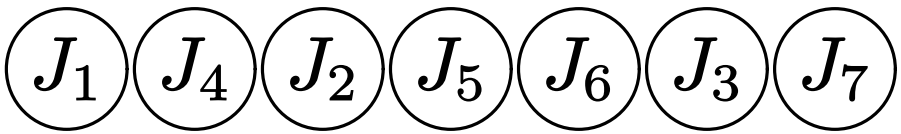

J_1 J_4 J_2 J_5 J_6 J_3 J_7

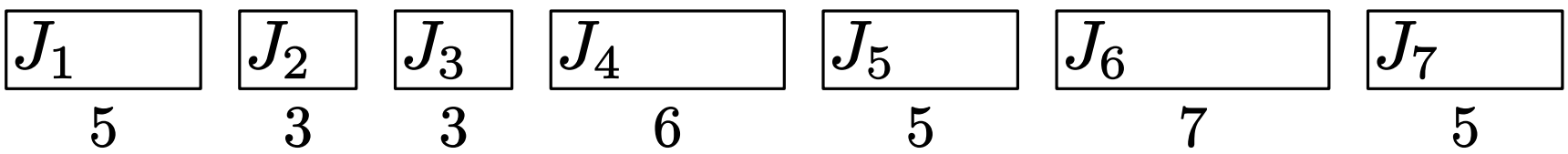

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

「後ろ」から決めていく

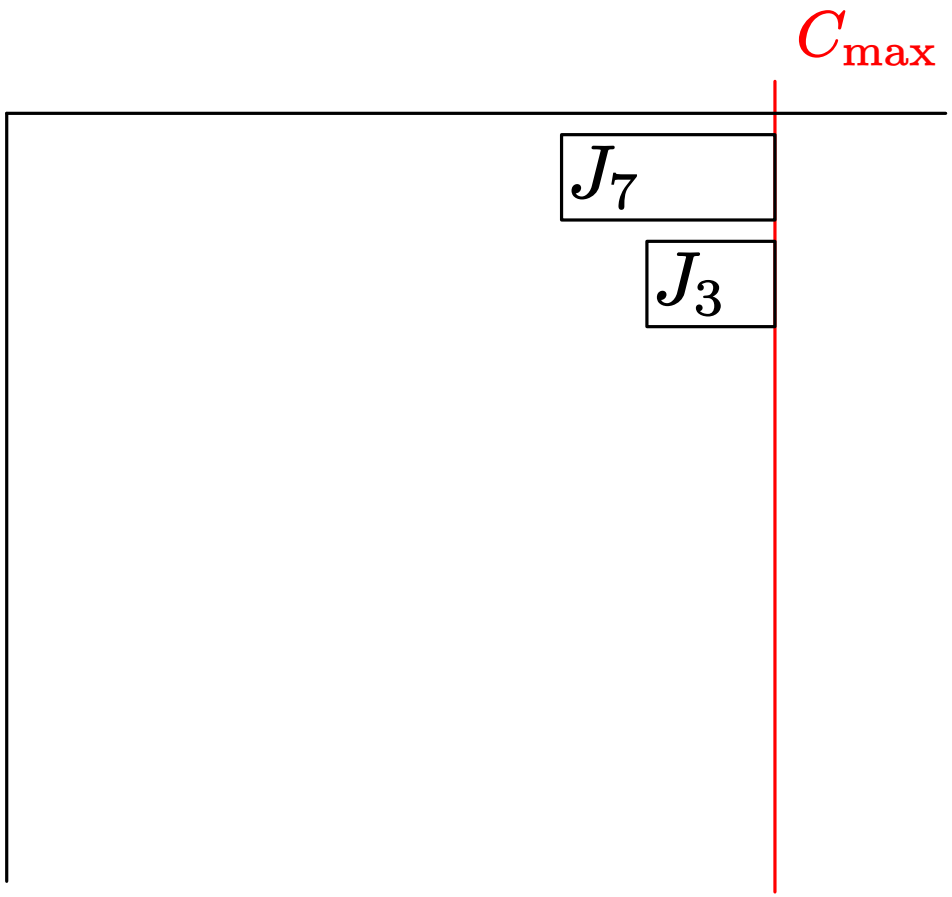

$C_{\max}$
J_7

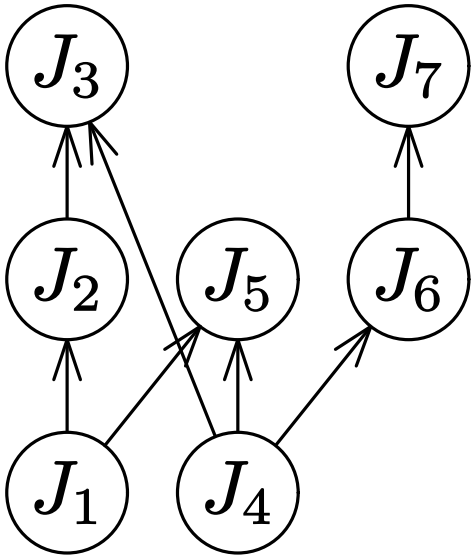

J_3
J_7
J_2
J_5
J_6
J_1
J_4

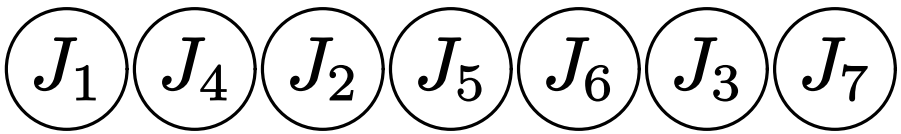

J_1 J_4 J_2 J_5 J_6 J_3 J_7

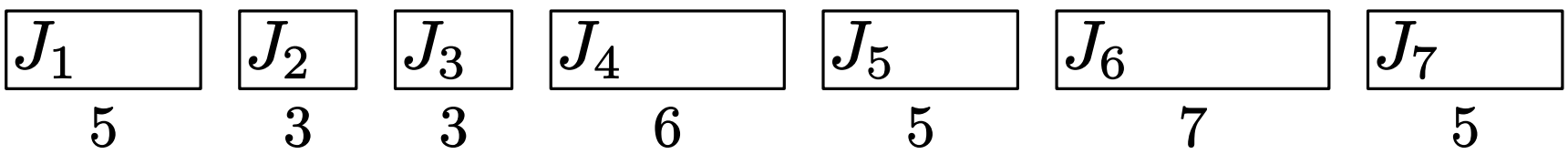

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

「後ろ」から決めていく

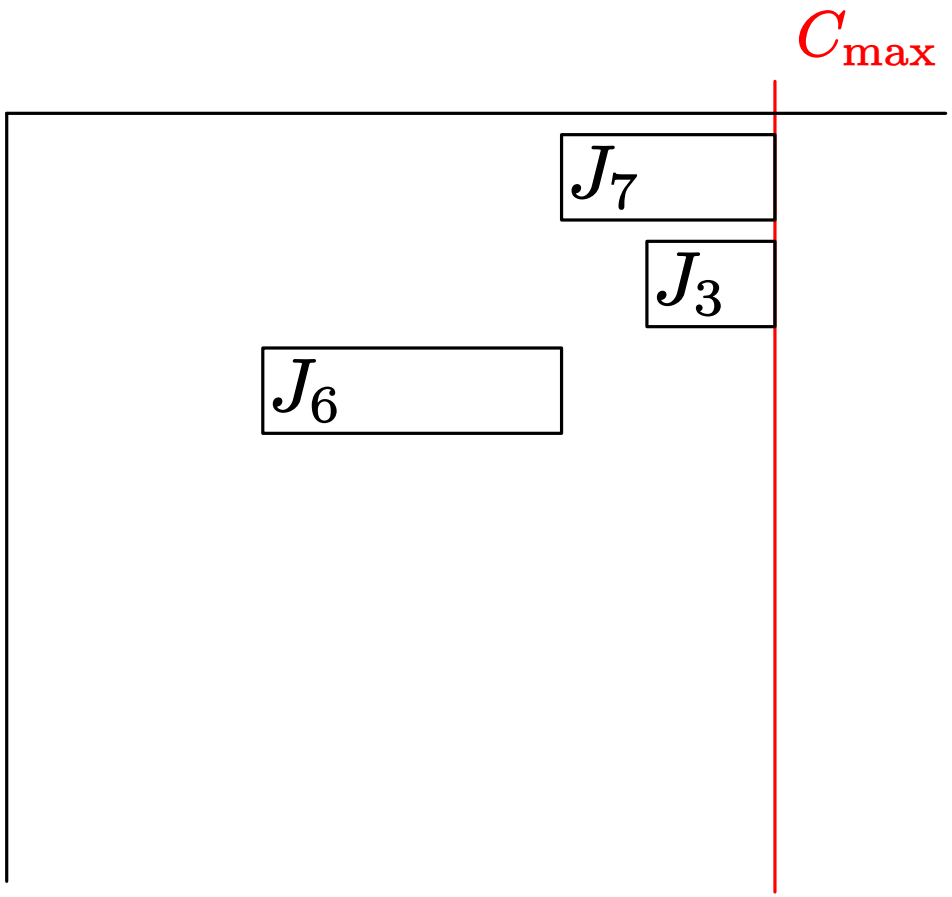

$C_{\max}$
J_7
J_3

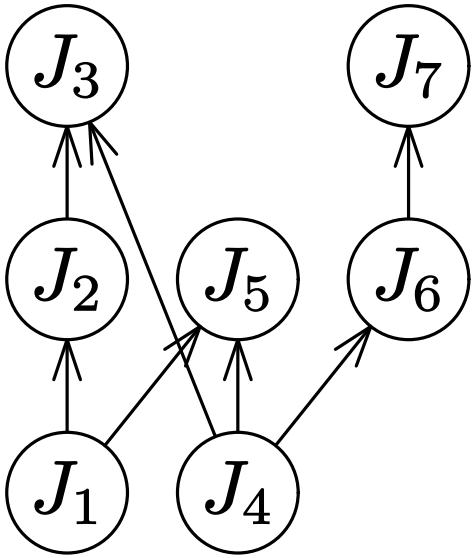

J_3
J_7
J_2
J_5
J_6
J_1
J_4

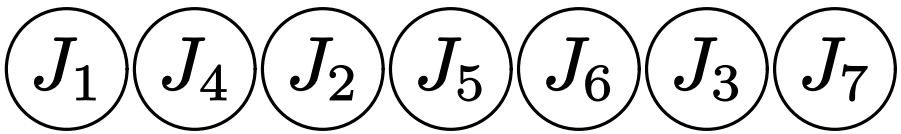

J_1 J_4 J_2 J_5 J_6 J_3 J_7

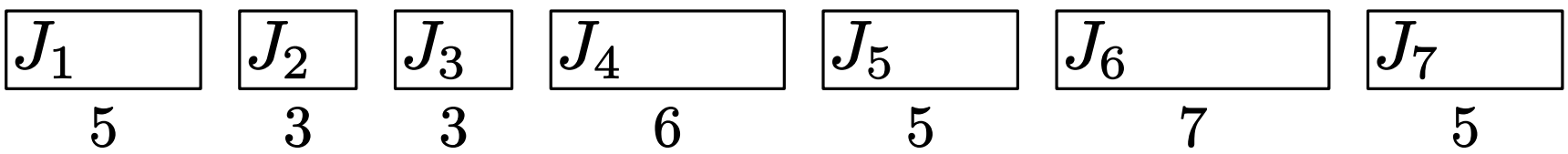

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

「後ろ」から決めていく

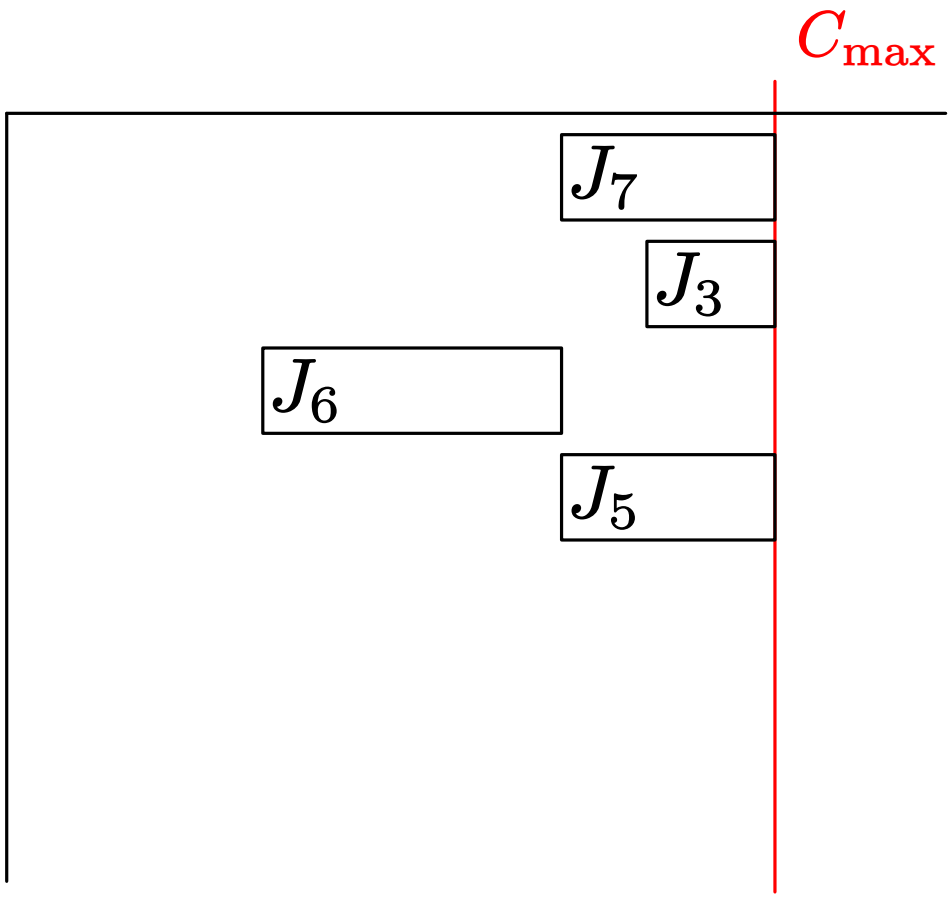

$C_{\max}$
J_7
J_3
J_6

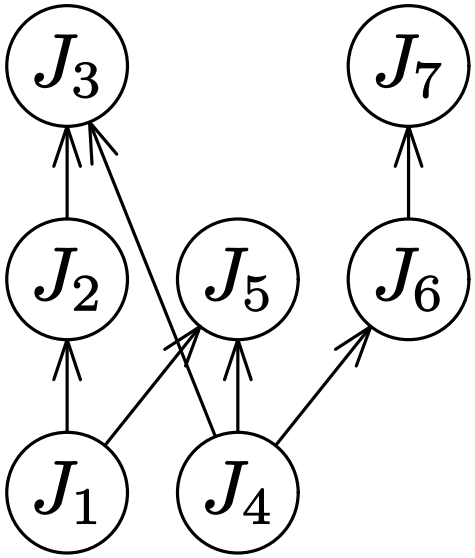

J_3
J_7
J_2
J_5
J_6
J_1
J_4

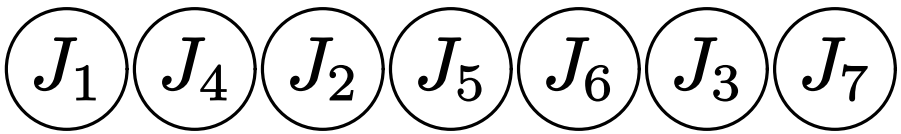

J_1 J_4 J_2 J_5 J_6 J_3 J_7

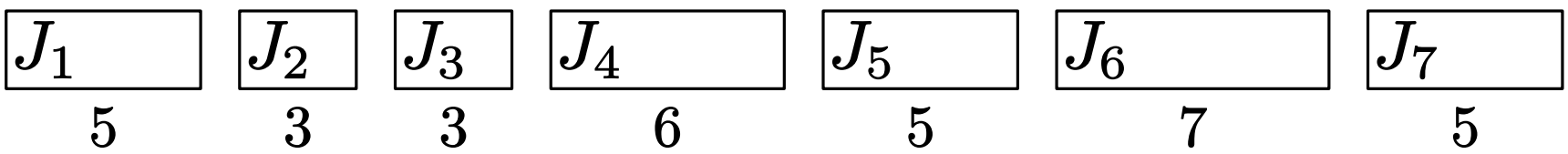

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

「後ろ」から決めていく

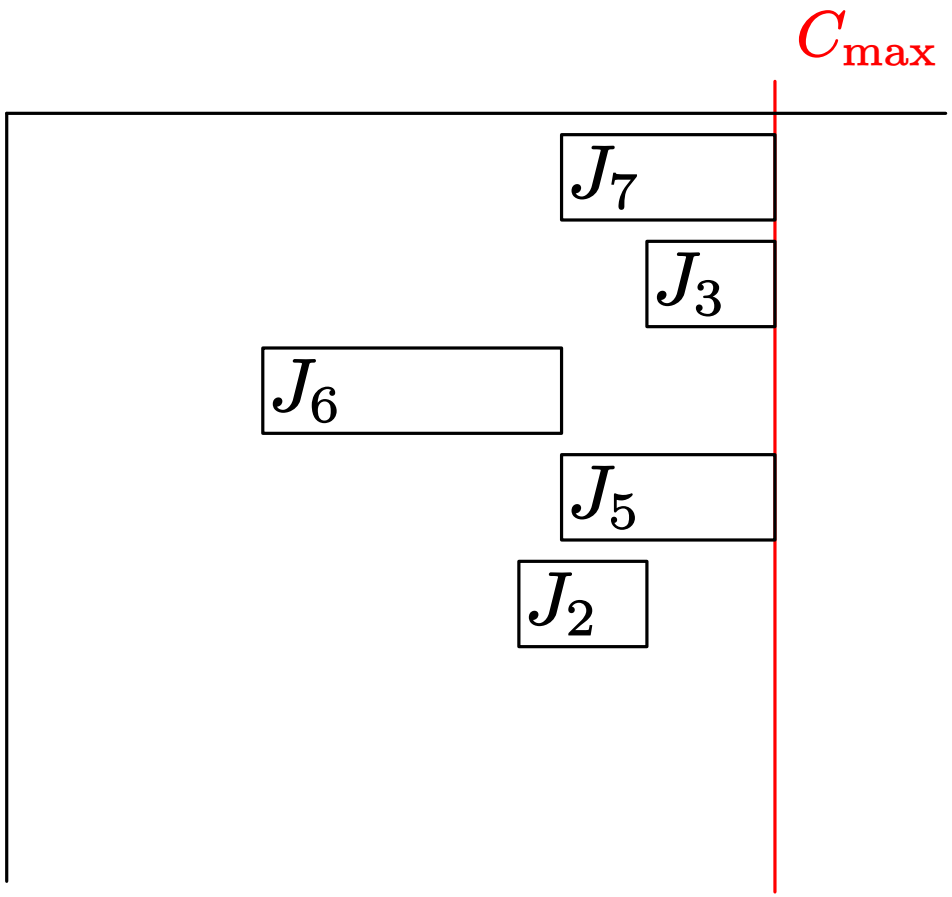

$C_{\max}$
J_7
J_3
J_6
J_5

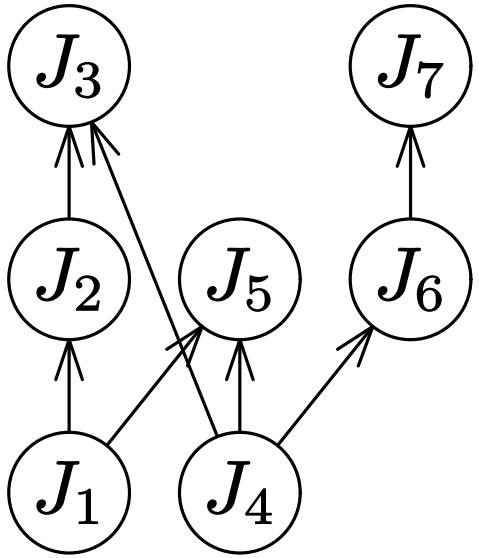

J_3
J_7
J_2
J_5
J_6
J_1
J_4

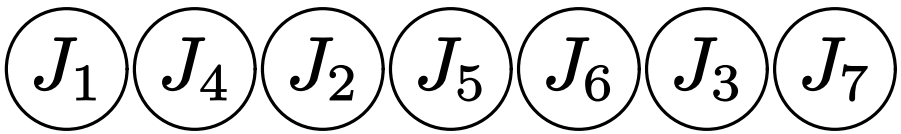

J_1 J_4 J_2 J_5 J_6 J_3 J_7

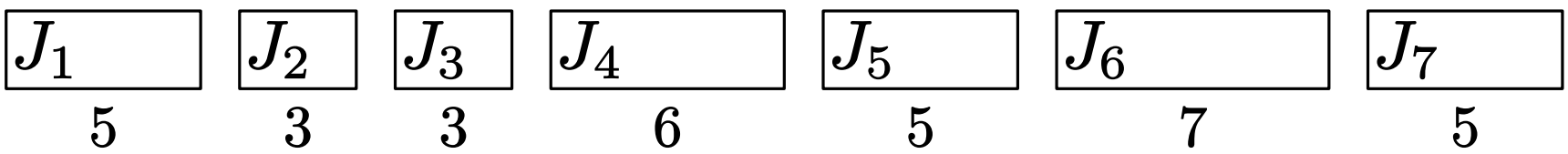

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

「後ろ」から決めていく

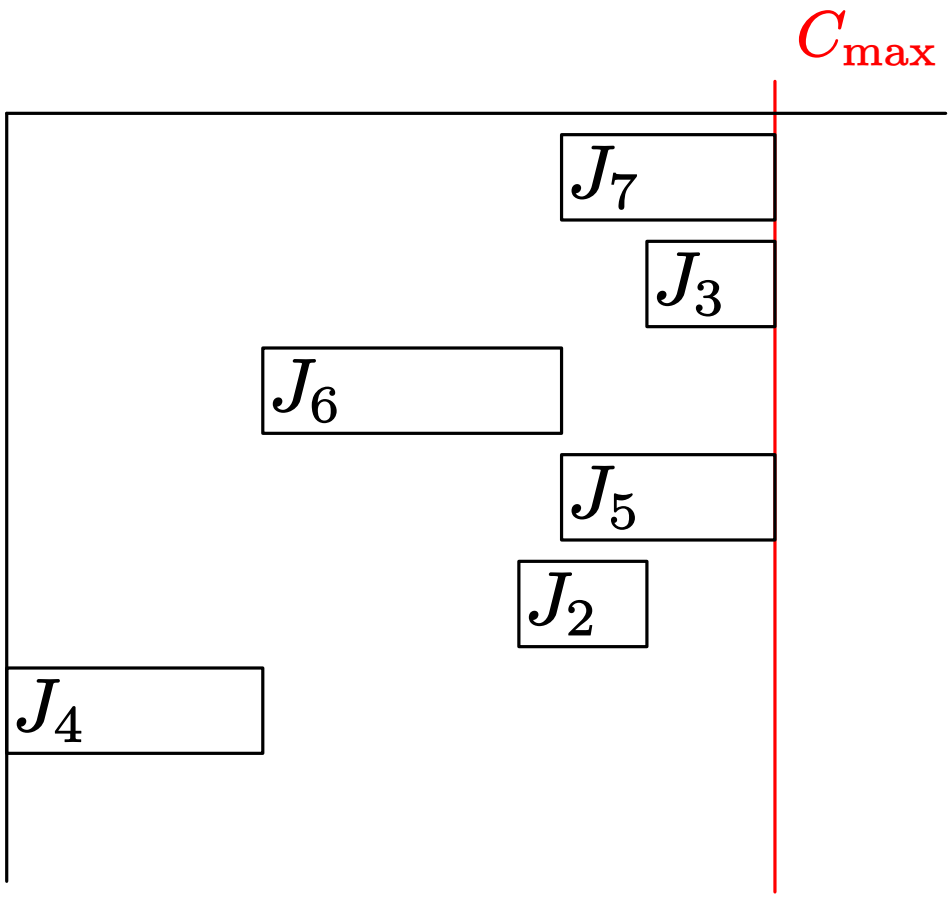

$C_{\max}$
J_7
J_3
J_6
J_5
J_2

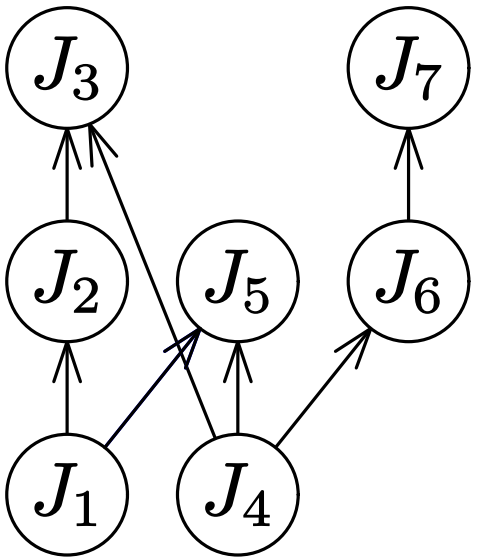

J_3
J_7
J_2
J_5
J_6
J_1
J_4

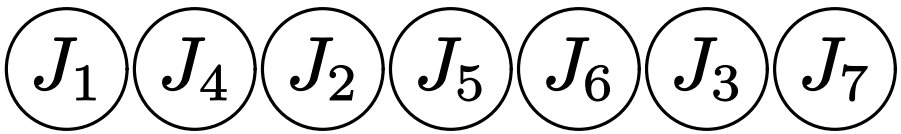

J_1 J_4 J_2 J_5 J_6 J_3 J_7

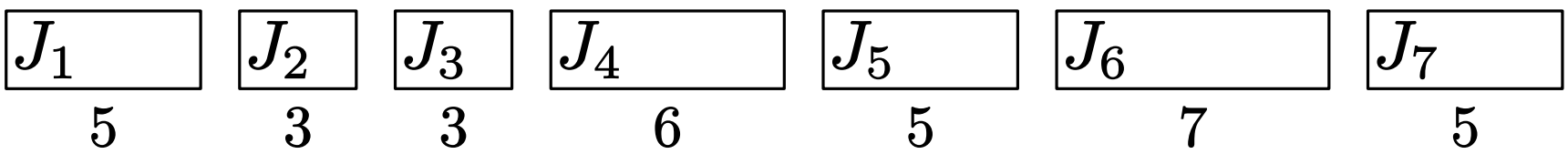

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

「後ろ」から決めていく

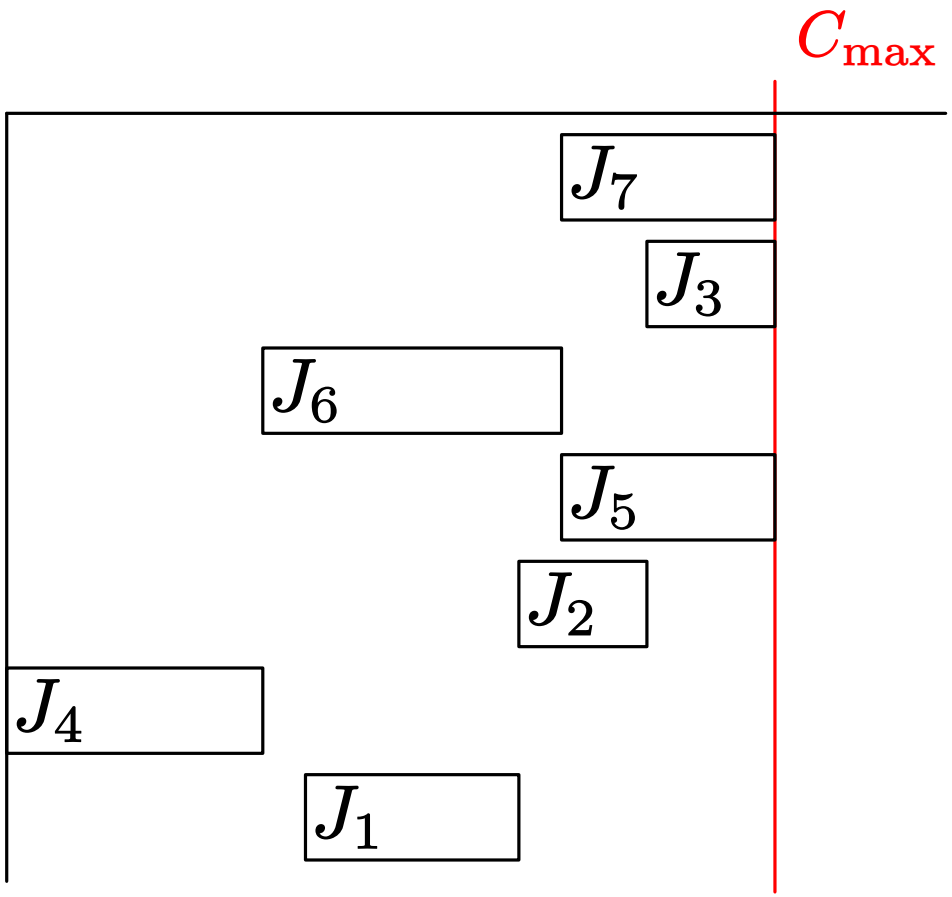

$C_{\max}$
J_7
J_3
J_6
J_5
J_2
J_4

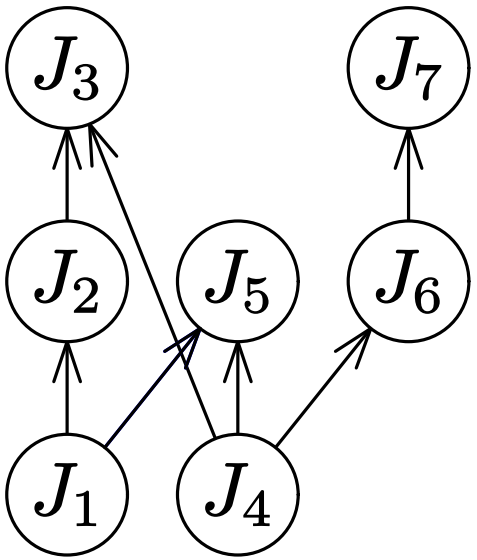

J_3
J_7
J_2
J_5
J_6
J_1
J_4

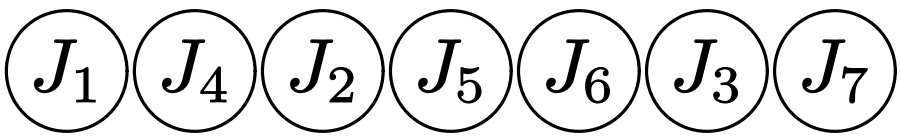

J_1 J_4 J_2 J_5 J_6 J_3 J_7

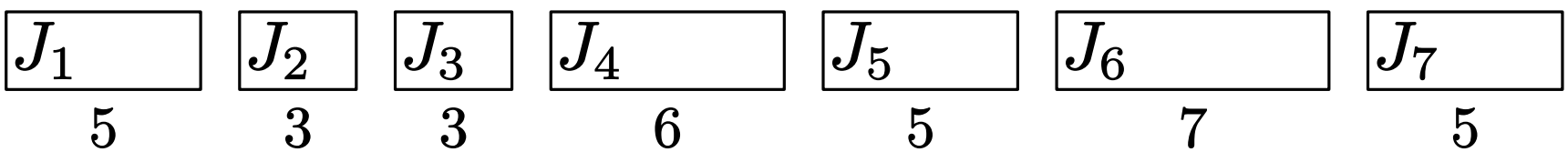

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

「後ろ」から決めていく

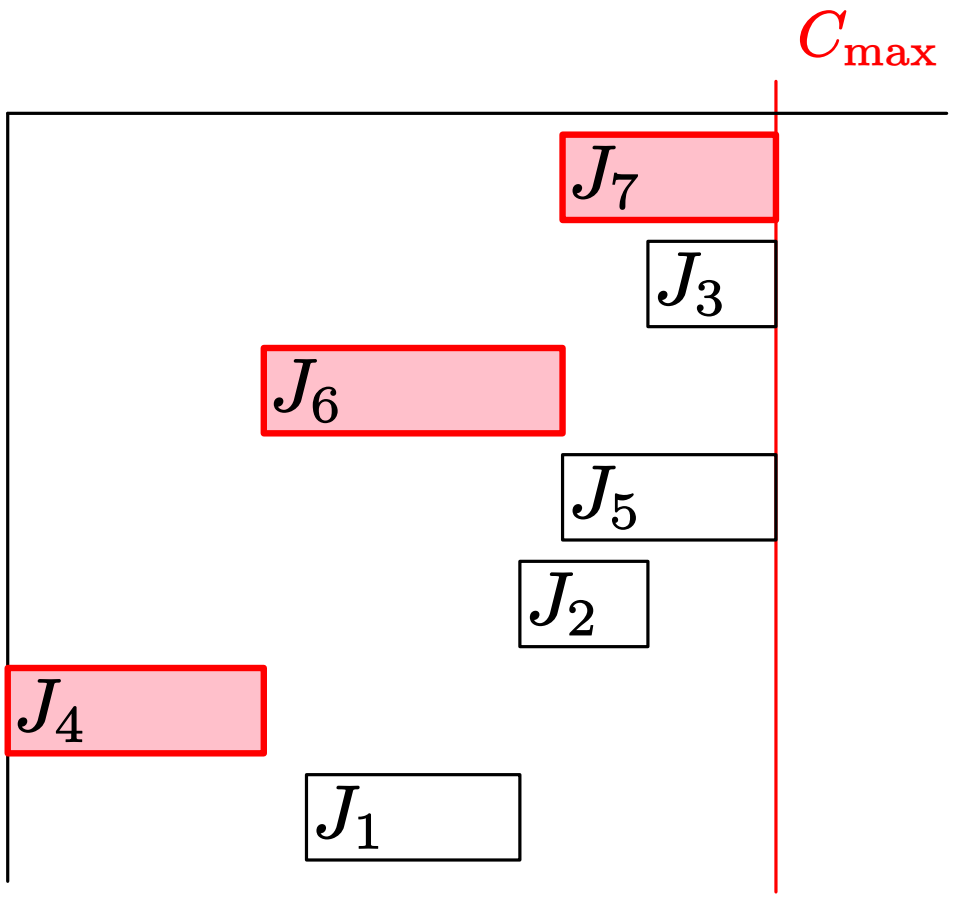

$C_{\max}$
J_7
J_3
J_6
J_5
J_2
J_4
J_1

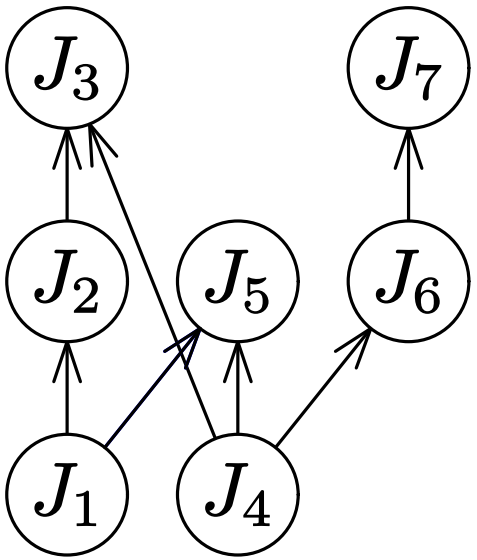

J_3
J_7
J_2
J_5
J_6
J_1
J_4

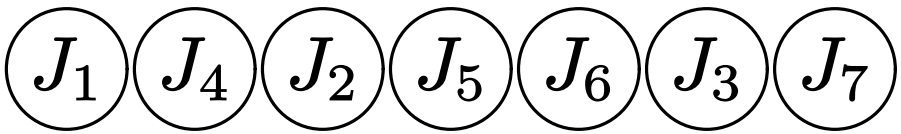

J_1 J_4 J_2 J_5 J_6 J_3 J_7

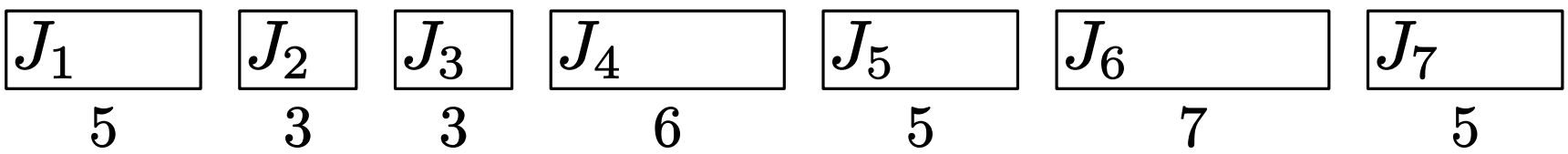

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

「後ろ」から決めていく

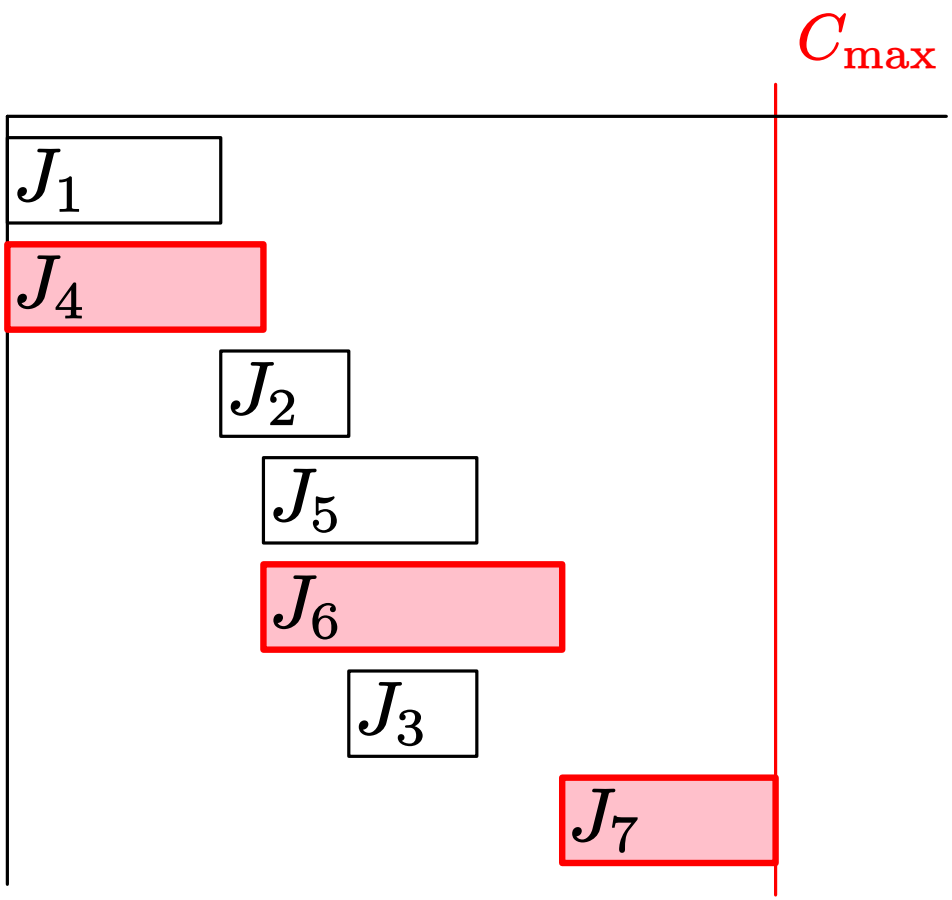

$C_{\max}$
J_7
J_3
J_6
J_5
J_2
J_4
J_1

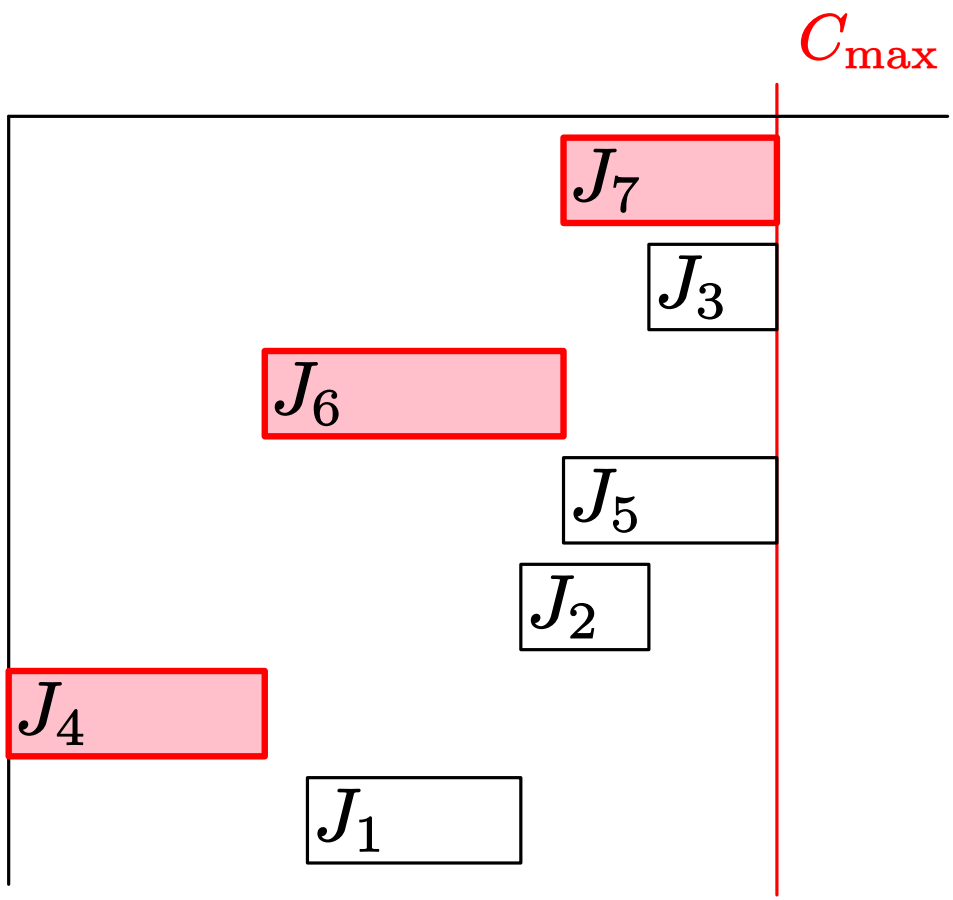

J_3
J_7
J_2
J_5
J_6
J_1
J_4


J_1 J_4 J_2 J_5 J_6 J_3 J_7

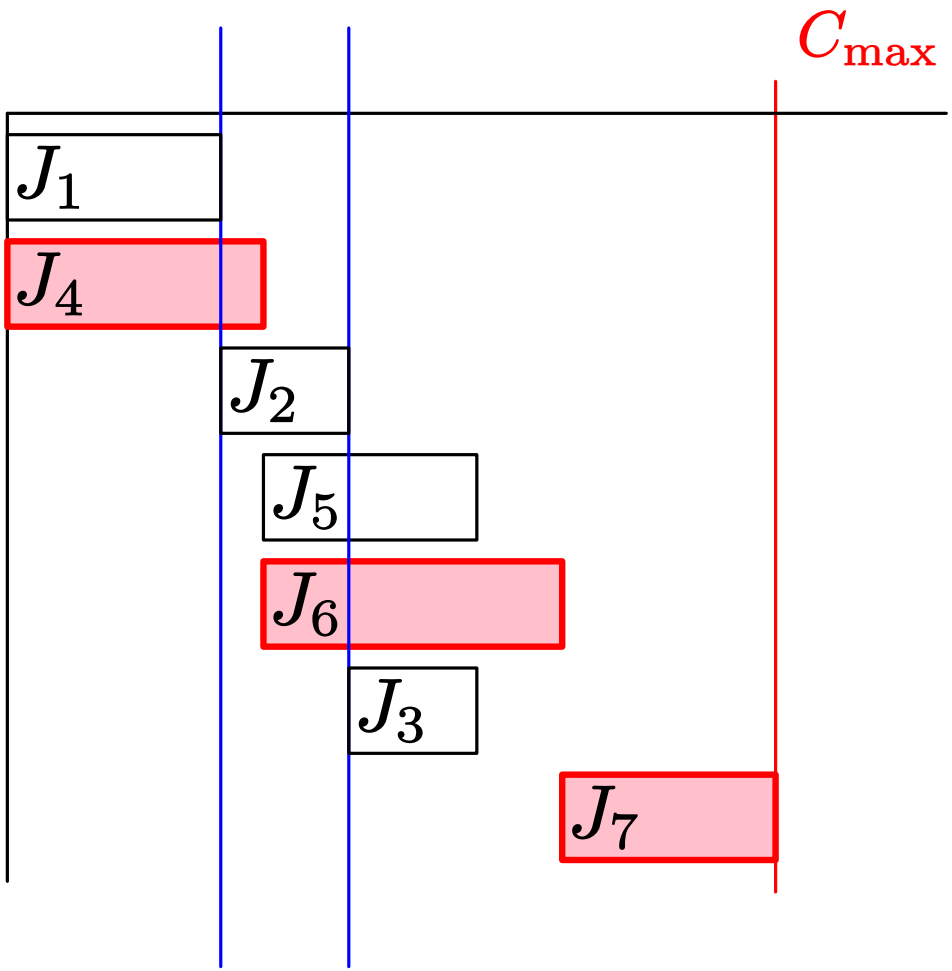

J_1 J_2 J_3 J_4 J_5 J_6 J_7
5 3 3 6 5 7 5

比較



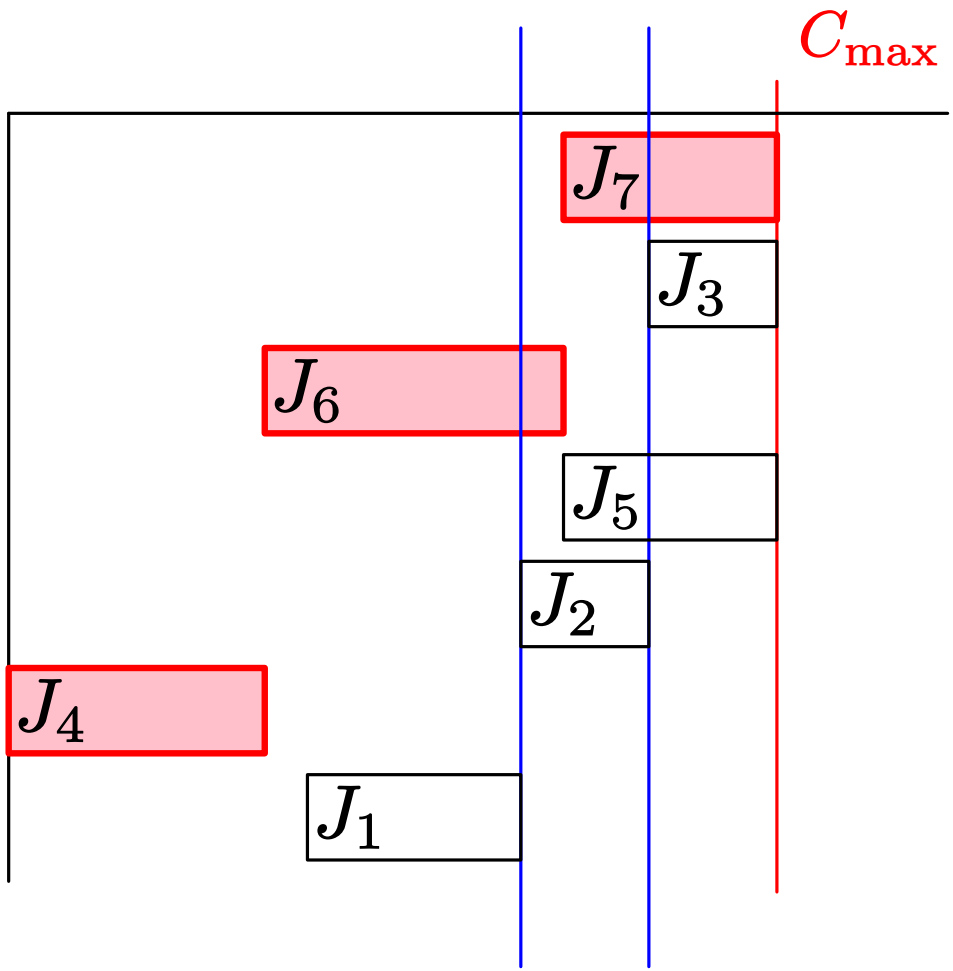


比較



J_2 の最早開始時刻

J_2 の最早完了時刻



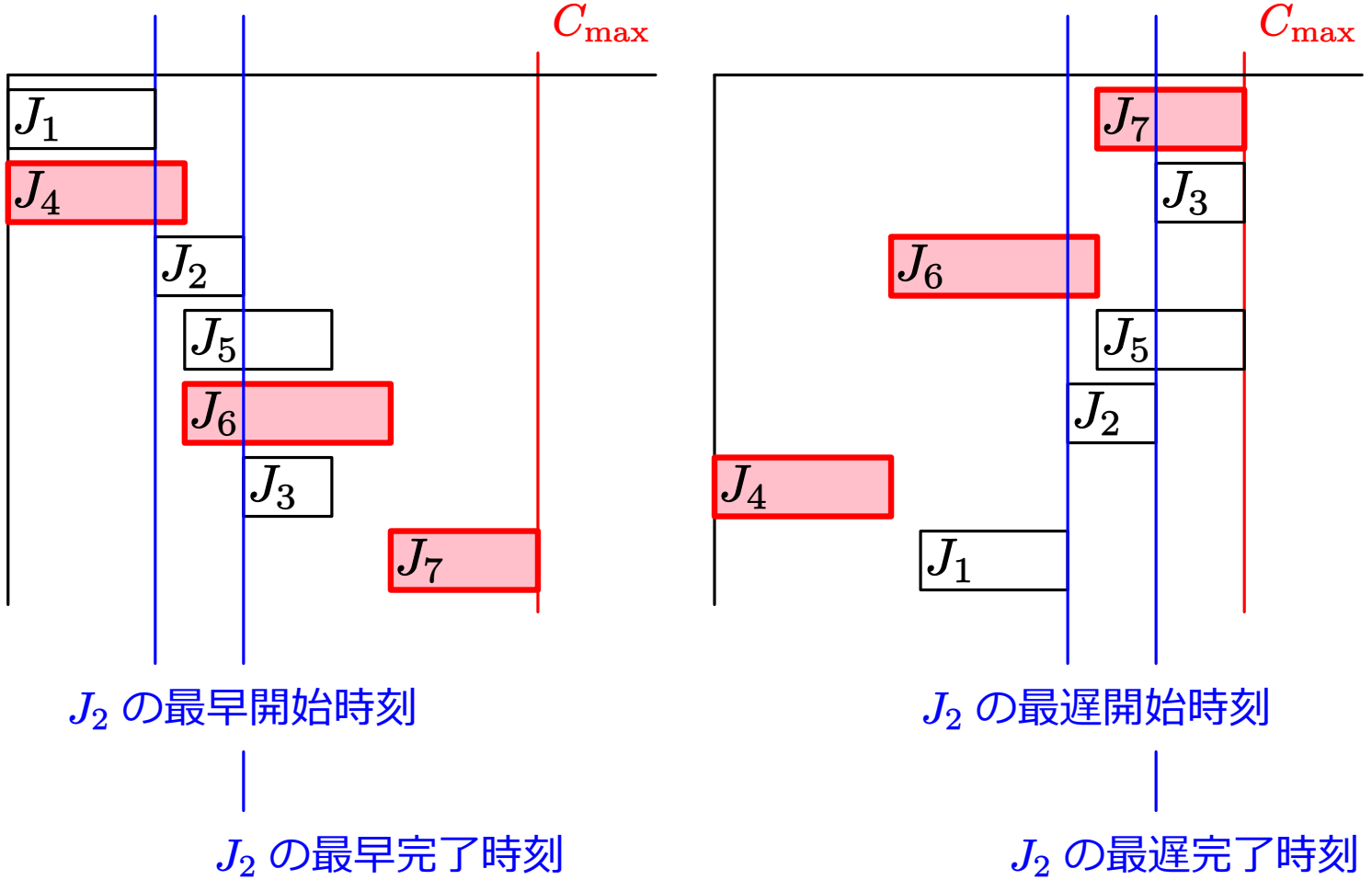
J_2 の最遅開始時刻

J_2 の最遅完了時刻

最早(開始|完了)時刻

最早開始時刻 (earliest starting date)：
これより早くジョブの処理を開始できない

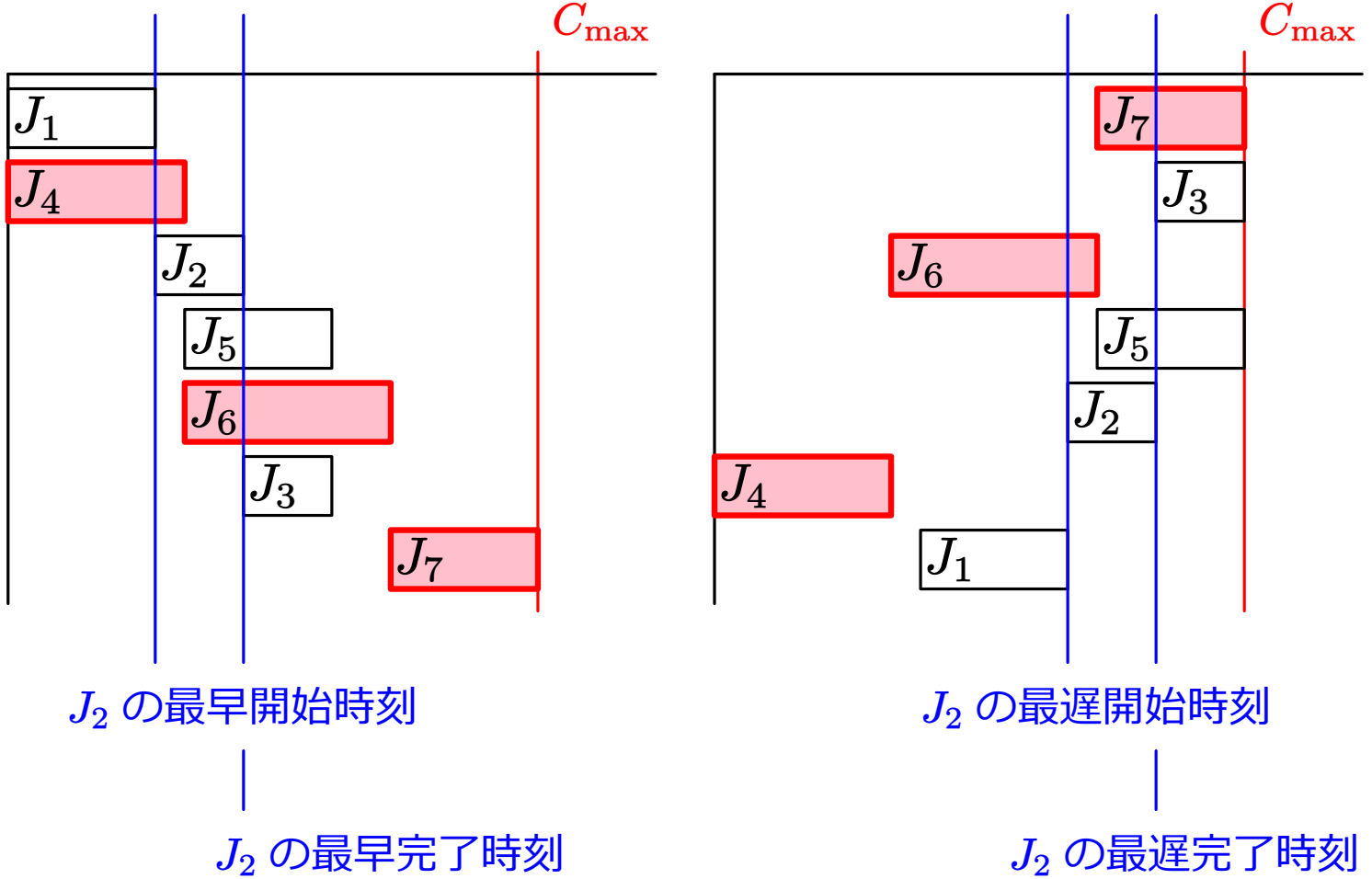
最早完了時刻 (earliest completion date, earliest finish date)：
これより早くジョブの処理を完了できない



最遅(開始|完了)時刻

最遅開始時刻 (latest starting date)：
これより遅く処理を開始すると，最大完了時刻が遅くなる

最遅完了時刻 (latest completion date, latest finish date)：
これより遅く処理を完了すると，最大完了時刻が遅くなる



まとめと次回の予告

目標

ジョブ・スケジューリングにおける
先行制約の扱い方 を学び, それを正しく使えるようになる

- 重要概念：トポロジカル順序，クリティカル・パス
- 主に対象とする問題
  - $1 \mid \mathrm{prec} \mid L_{\max}$
  - $1 \mid \mathrm{prec} \mid \sum U_j$
  - $P\infty \mid \mathrm{prec} \mid C_{\max}$

次回の予告

先行制約 + 一様並列機械 (機械数は有限) における
スケジューリング

付録

1. 半順序集合

半順序集合

定義：半順序集合 (partially ordered set, poset)

半順序集合 とは次を満たす組 $\mathcal{P} = (X, \preceq)$

- X は集合
- $\preceq$ は X 上の半順序 (partial order)
  - $x \preceq x$ (反射性)
  - $x \preceq y$ かつ $y \preceq x$ ならば $x = y$ (反対称性)
  - $x \preceq y$ かつ $y \preceq z$ ならば, $x \preceq z$ (推移性)

	a	b	c	d	e	f
a	$\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
b	$\succeq$	$\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
c	$\not\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
d	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$	$\not\preceq$	$\preceq$
e	$\not\preceq$	$\not\preceq$	$\succeq$	$\not\preceq$	$\preceq$	$\preceq$
f	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$

$$X = \{a, b, c, d, e, f\}$$

半順序集合

定義：半順序集合 (partially ordered set, poset)

半順序集合 とは次を満たす組 $\mathcal{P} = (X, \preceq)$

- X は集合
- $\preceq$ は X 上の半順序 (partial order)
  - $x \preceq x$ (反射性)
  - $x \preceq y$ かつ $y \preceq x$ ならば $x = y$ (反対称性)
  - $x \preceq y$ かつ $y \preceq z$ ならば，$x \preceq z$ (推移性)

	a	b	c	d	e	f
a	$\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
b	$\preceq$	$\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
c	$\not\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
d	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$	$\not\preceq$	$\preceq$
e	$\not\preceq$	$\not\preceq$	$\preceq$	$\not\preceq$	$\preceq$	$\preceq$
f	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$

$$X = \{a, b, c, d, e, f\}$$

半順序集合

定義：半順序集合 (partially ordered set, poset)

半順序集合 とは次を満たす組 $\mathcal{P} = (X, \preceq)$

- X は集合
- $\preceq$ は X 上の半順序 (partial order)
  - $x \preceq x$ (反射性)
  - $x \preceq y$ かつ $y \preceq x$ ならば $x = y$ (反対称性)
  - $x \preceq y$ かつ $y \preceq z$ ならば, $x \preceq z$ (推移性)

	a	b	c	d	e	f
a	$\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
b	$\preceq$	$\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
c	$\not\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
d	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$	$\not\preceq$	$\preceq$
e	$\not\preceq$	$\not\preceq$	$\preceq$	$\not\preceq$	$\preceq$	$\preceq$
f	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$

$$X = \{a, b, c, d, e, f\}$$

半順序集合

定義：半順序集合 (partially ordered set, poset)

半順序集合 とは次を満たす組 $\mathcal{P} = (X, \preceq)$

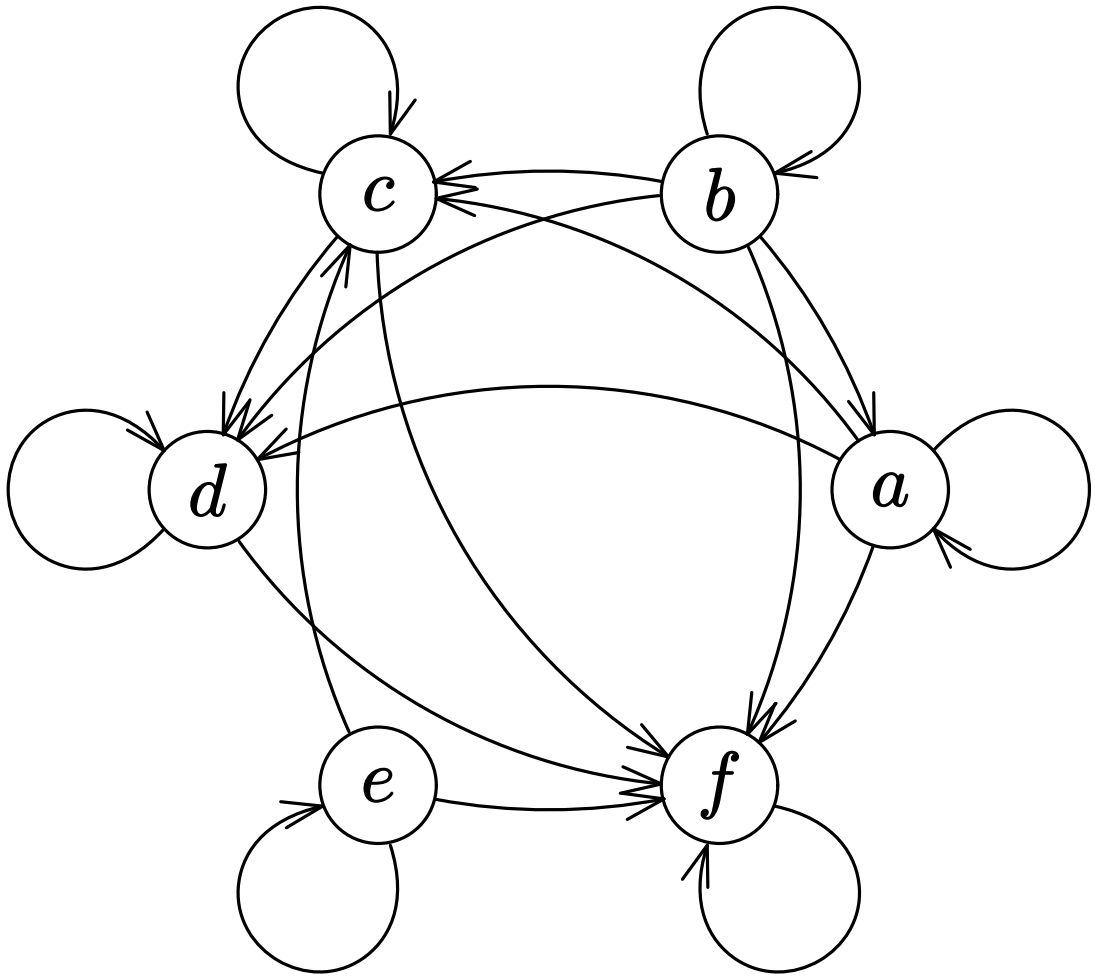
- X は集合
- $\preceq$ は X 上の半順序 (partial order)
  - $x \preceq x$ (反射性)
  - $x \preceq y$ かつ $y \preceq x$ ならば $x = y$ (反対称性)
  - $x \preceq y$ かつ $y \preceq z$ ならば, $x \preceq z$ (推移性)

	a	b	c	d	e	f
a	$\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
b	$\succeq$	$\preceq$	$\succeq$	$\preceq$	$\not\preceq$	$\succeq$
c	$\not\preceq$	$\not\preceq$	$\preceq$	$\succeq$	$\not\preceq$	$\succeq$
d	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$	$\not\preceq$	$\succeq$
e	$\not\preceq$	$\not\preceq$	$\succeq$	$\not\preceq$	$\preceq$	$\succeq$
f	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$

$$X = \{a, b, c, d, e, f\}$$

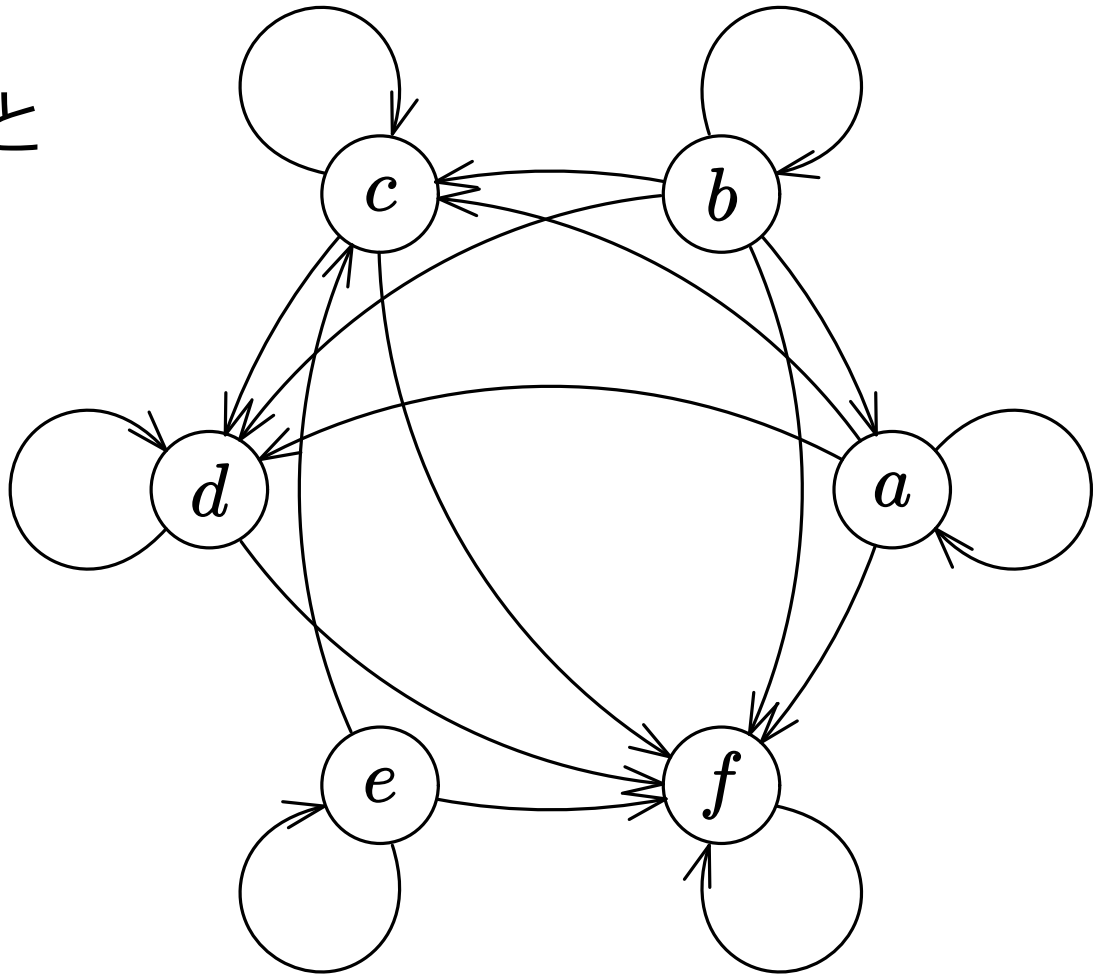
半順序集合から得られる有向グラフ

	a	b	c	d	e	f
a	$\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
b	$\preceq$	$\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
c	$\not\preceq$	$\not\preceq$	$\preceq$	$\preceq$	$\not\preceq$	$\preceq$
d	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$	$\not\preceq$	$\preceq$
e	$\not\preceq$	$\not\preceq$	$\preceq$	$\not\preceq$	$\preceq$	$\preceq$
f	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\not\preceq$	$\preceq$



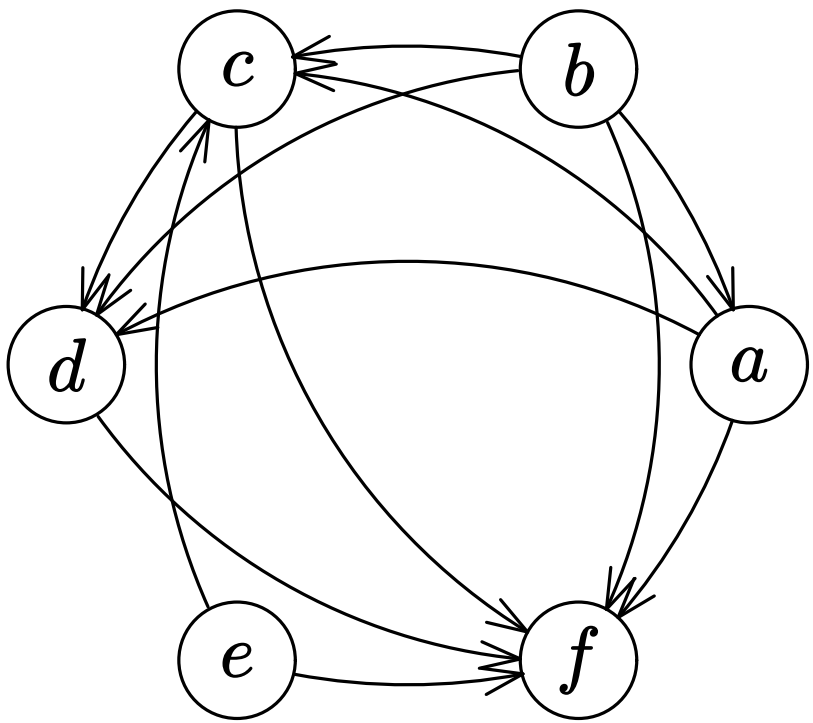
冗長な情報を取り除く

半順序集合から得られたと
分かっていたら・・・



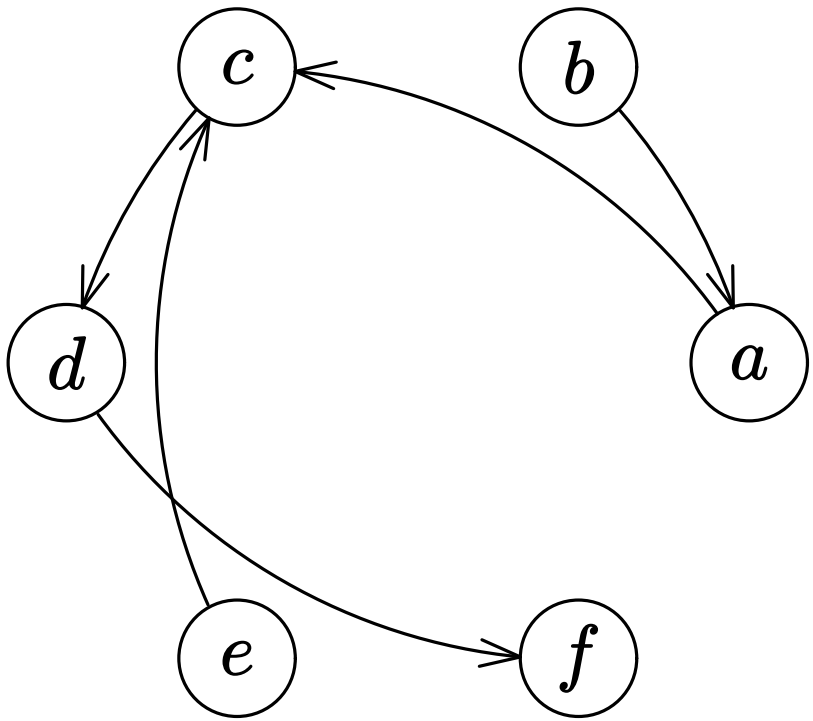
冗長な情報を取り除く

半順序集合から得られたと
分かっていたら · · ·



冗長な情報を取り除く

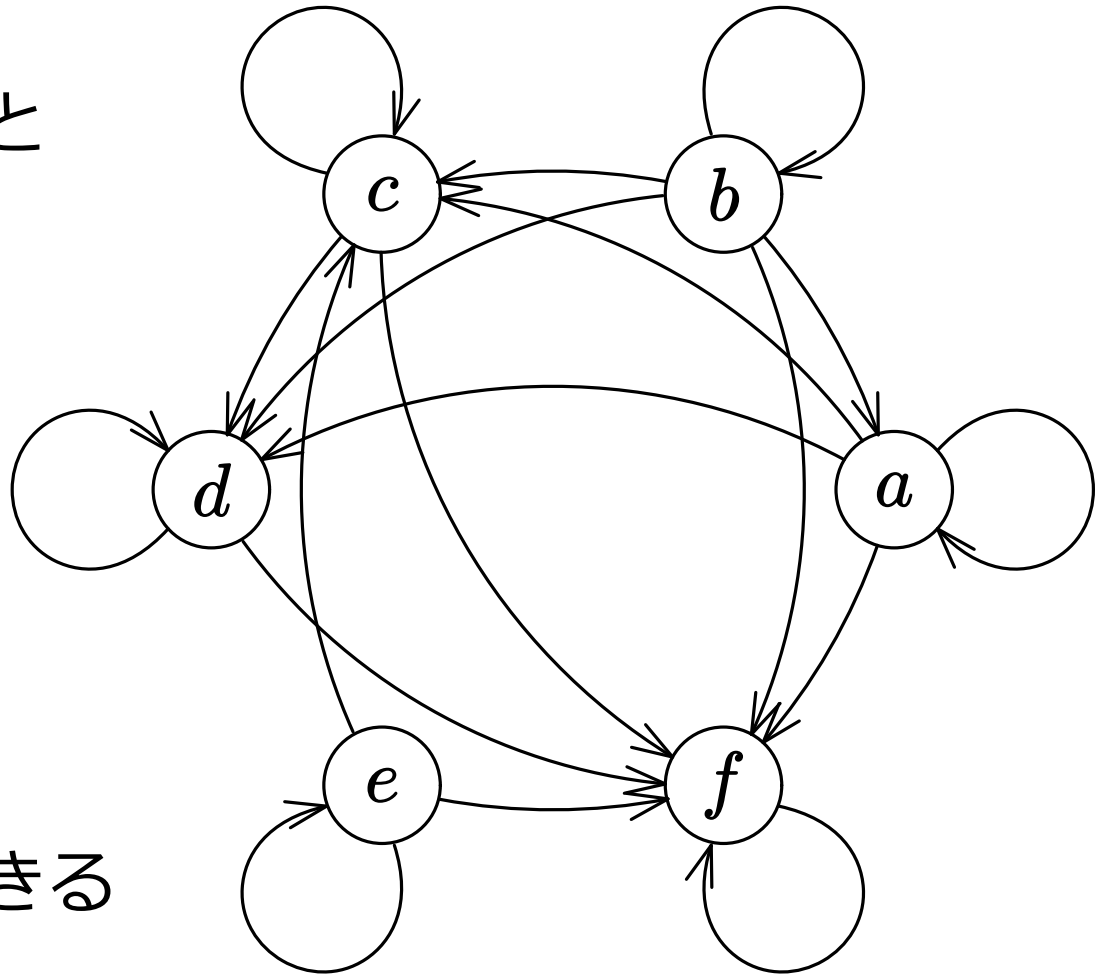
半順序集合から得られたと
分かっていたら・・・



このグラフから
元の半順序集合が復元できる

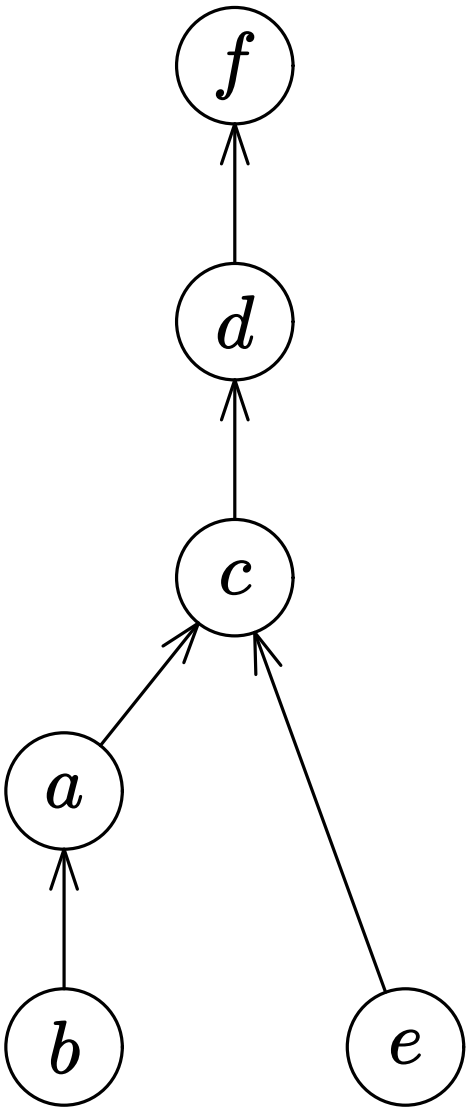
冗長な情報を取り除く

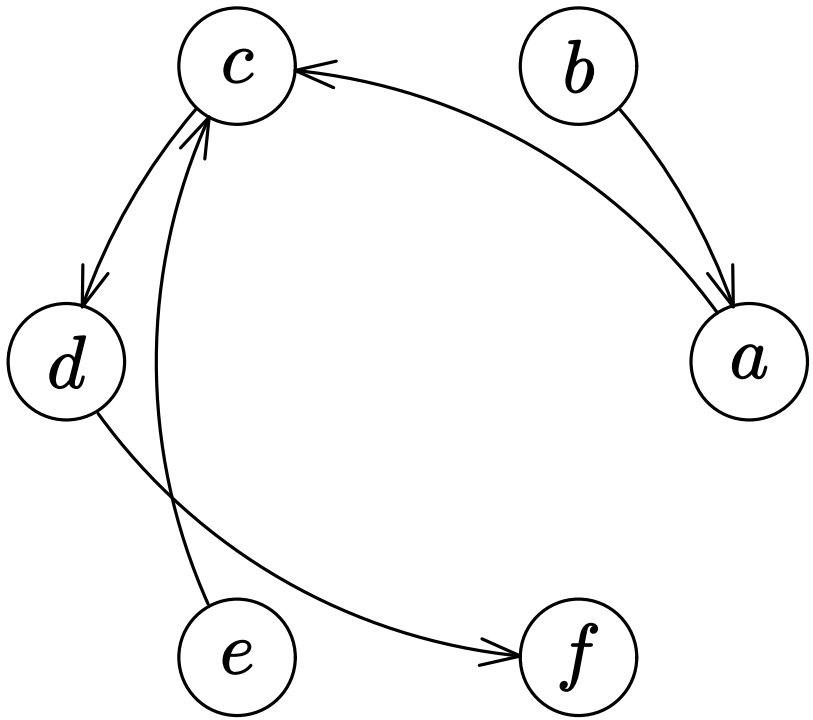
半順序集合から得られたと
分かっていたら・・・



このグラフから
元の半順序集合が復元できる

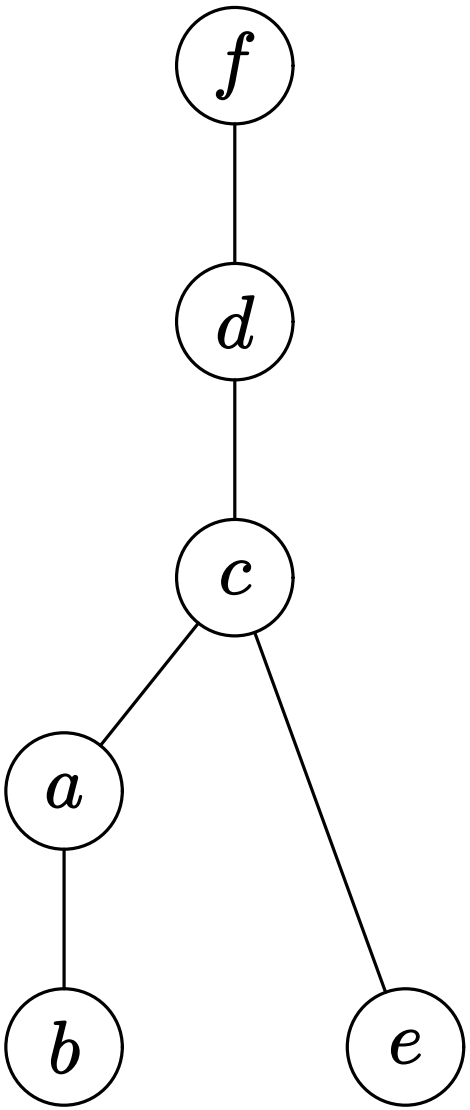
半順序集合のハッセ図

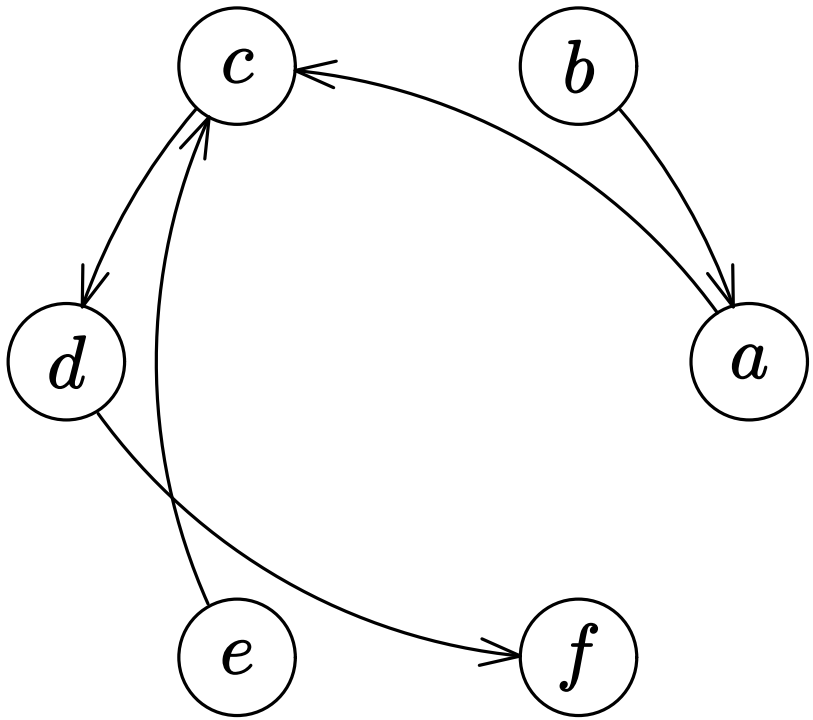




ハッセ図 (Hasse diagram)

半順序集合のハッセ図





ハッセ図 (Hasse diagram)