離散最適化基礎論　第 8 回
数値が関わる問題 (1) : 2 分割問題

岡本 吉央
okamotoy@uec.ac.jp

電気通信大学

2019 年 12 月 3 日

最終更新：2019 年 12 月 2 日　11:55

スケジュール 前半

スケジュール 後半 (予定)

注意：予定の変更もありうる

今日の目標

今日の目標

- 数値が関わる問題に関する基礎概念を理解する
  - 数値の入力法：2 進法，1 進法
  - 強多項式時間，弱多項式時間，擬多項式時間
  - 強 NP 完全，弱 NP 完全
- 数値が関わる問題の中で最も基礎的な NP 完全問題を理解する
  - 2 分割問題
- 数値が関わる問題について，NP 完全性の証明を行う
  - 部分和問題，ナップサック問題，スケジューリング問題

目次

2 分割問題

2 分割問題 (2-partition problem，分割問題とも呼ばれる)

- 入力：n 個の自然数の集合 $A = \{a_1, a_2, \ldots, a_n\} \subseteq \mathbb{N}$
- 出力：Yes か No
- 条件：$\sum_{i \in X} a_i = \sum_{i \not\in X} a_i$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在する $\Rightarrow$ Yes
  $\sum_{i \in X} a_i = \sum_{i \not\in X} a_i$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在しない $\Rightarrow$ No

a_1	3
a_2	7
a_3	2
a_4	4
a_5	3
a_6	2
a_7	1
和	22

これは Yes 入力：$X = \{1, 2, 7\}$ が解

- $\sum_{i \in X} a_i = a_1 + a_2 + a_7 = 3 + 7 + 1 = 11$
- $\sum_{i \not\in X} a_i = a_3 + a_4 + a_5 + a_6 = 2 + 4 + 3 + 2 = 11$

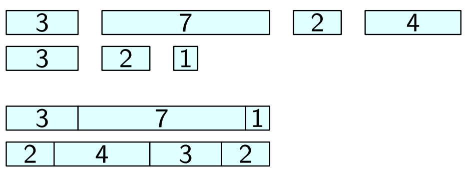
注：2 分割問題は「同じ和をもつ 2 つの部分」に分割する

2 分割問題

2 分割問題 (2-partition problem，分割問題とも呼ばれる)

- 入力：n 個の自然数の集合 $A = \{a_1, a_2, \ldots, a_n\} \subseteq \mathbb{N}$
- 出力：Yes か No
- 条件：$\sum_{i \in X} a_i = \sum_{i \not\in X} a_i$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在する $\Rightarrow$ Yes
  $\sum_{i \in X} a_i = \sum_{i \not\in X} a_i$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在しない $\Rightarrow$ No

a_1	3
a_2	7
a_3	2
a_4	4
a_5	3
a_6	2
a_7	1
和	22



2 分割問題の NP 完全性

定理 (Karp '72)

2 分割問題は NP 完全

証明はややこしいので，省略

ここで気にしたいこと

- 数値はどのように入力するのか？
- 数値の入力法によって，問題の難しさは変わるのか？

自然数の入力法

自然数の符号化法：2 種類

- 1 進符号化 (unary encoding)
- 2 進符号化 (binary encoding) ←普通の入力法

10 進	1 進符号化	2 進符号化
0	0	0
1	10	1
2	110	10
3	1110	11
4	11110	100
5	111110	101
6	1111110	110
7	11111110	111
8	111111110	1000

自然数 a を符号化するとき，

- 1 進符号化のサイズ $= O(a)$
- 2 進符号化のサイズ $= O(\log a)$

つまり，2 つの符号化法の間で
サイズが指数関数的に異なる

弱多項式時間アルゴリズムと擬多項式時間アルゴリズム

数値を入力 (の一部) とする問題に対して

- 弱多項式時間アルゴリズム (weakly polynomial-time algorithm) とは，
入力数値が 2 進符号化されるとき，
多項式時間で実行できるアルゴリズム
- 擬多項式時間アルゴリズム (pseudo-polynomial-time algorithm) とは，
入力数値が 1 進符号化されるとき，
多項式時間で実行できるアルゴリズム

例：入力数値が自然数 $a_1, a_2, \ldots, a_n$ であるとき，

- 最悪時実行時間 $= O\left(\sum_{i=1}^{n} \log a_i\right)$ ： 弱多項式時間アルゴリズム
- 最悪時実行時間 $= O\left(\sum_{i=1}^{n} a_i\right)$ ： 擬多項式時間アルゴリズム
2 進符号化だと，指数時間

強多項式時間アルゴリズム

数値を入力 (の一部) とする問題に対して

- 強多項式時間アルゴリズム (strongly polynomial-time algorithm)
とは，数値の総数の多項式時間で実行できるアルゴリズム

例：入力数値が自然数 $a_1, a_2, \ldots, a_n$ であるとき，

- 最悪時実行時間 $= O\left(\sum_{i=1}^{n} \log a_i\right)$ ： 弱多項式時間アルゴリズム
- 最悪時実行時間 $= O\left(n^2\right)$ ： 強多項式時間アルゴリズム

3 つの多項式時間アルゴリズムの比較

例：入力数値が自然数 $a_1, a_2, \ldots, a_n$ であるとき，

- 最悪時実行時間 $= O\left(n^2\right)$ ： 強多項式時間アルゴリズム
- 最悪時実行時間 $= O\left(\sum_{i=1}^{n} \log a_i\right)$ ： 弱多項式時間アルゴリズム
- 最悪時実行時間 $= O\left(\sum_{i=1}^{n} a_i\right)$ ： 擬多項式時間アルゴリズム

普通の入力法 (つまり，2 進符号化) において

- 強多項式時間アルゴリズム，弱多項式時間アルゴリズムは
多項式時間アルゴリズム
- 擬多項式時間アルゴリズムは
指数時間アルゴリズム

速さの直感：強多項式時間 $>$ 弱多項式時間 $>$ 擬多項式時間

弱 NP 完全性と強 NP 完全性

数値を入力 (の一部) とする問題 P に対して

- 問題 P が弱 NP 完全 (weakly NP-complete) であるとは，
入力数値が 2 進符号化されるときに，NP 完全であること
- 問題 P が強 NP 完全 (strongly NP-complete) であるとは，
入力数値が 1 進符号化されるときに，NP 完全であること

$\mathrm{P} \neq \mathrm{NP}$ であるとき，

- 問題 P が弱 NP 完全である $\Rightarrow$
  - P を解く弱多項式時間アルゴリズムは存在しない
  - (P を解く擬多項式時間アルゴリズムは存在するかもしれない)
- 問題 P が強 NP 完全である $\Rightarrow$
  - P を解く擬多項式時間アルゴリズムは存在しない
  - (つまり，P を解く弱多項式時間アルゴリズムも存在しない)

強，弱，擬

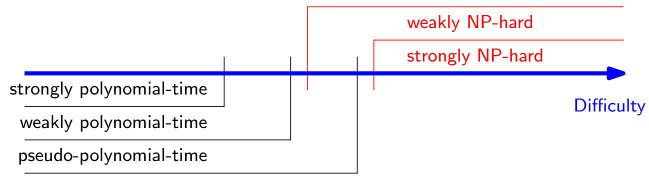
アルゴリズムの速さに対する直感

強多項式時間 $>$ 弱多項式時間 $>$ 擬多項式時間

問題の難しさに対する直感

弱 NP 完全 $<$ 強 NP 完全

$\mathrm{P} \neq \mathrm{NP}$ であるとき



2 分割問題：再掲

2 分割問題 (2-partition problem，分割問題とも呼ばれる)

- 入力：n 個の自然数の集合 $A = \{a_1, a_2, \ldots, a_n\} \subseteq \mathbb{N}$
- 出力：Yes か No
- 条件：$\sum_{i \in X} a_i = \sum_{i \notin X} a_i$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在する $\Rightarrow$ Yes
$\sum_{i \in X} a_i = \sum_{i \notin X} a_i$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在しない $\Rightarrow$ No

a_1	3
a_2	7
a_3	2
a_4	4
a_5	3
a_6	2
a_7	1
和	22

これは Yes 入力：$X = \{1, 2, 7\}$ が解

- $\sum_{i \in X} a_i = a_1 + a_2 + a_7 = 3 + 7 + 1 = 11$
- $\sum_{i \notin X} a_i = a_3 + a_4 + a_5 + a_6 = 2 + 4 + 3 + 2 = 11$

注：2 分割問題は「同じ和をもつ 2 つの部分」に分割する

2 分割問題の弱 NP 完全性

定理 (Karp '72)

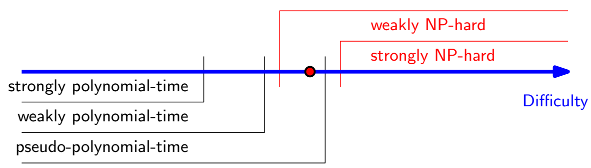
2 分割問題は弱 NP 完全

証明はややこしいので，省略

事実 (Bellman '56, Dantzig '57)

2 分割問題を解く擬多項式時間アルゴリズムが存在する

- 最悪時実行時間 $= O\left(n \sum_{i=1}^{n} a_i\right)$ (動的計画法の典型例)



目次

1. 数値が関わる問題：2 分割問題
2. 2 分割問題に似た問題
3. スケジューリング問題
4. 今日のまとめ と 次回の予告

部分和問題

部分和問題 (subset sum problem)

- 入力：n 個の自然数の集合 $A = \{a_1, a_2, \ldots, a_n\} \subseteq \mathbb{N}$, 自然数 $t \in \mathbb{N}$
- 出力：Yes か No
- 条件：$\sum_{i \in X} a_i = t$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在する $\Rightarrow$ Yes
  $\sum_{i \in X} a_i = t$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在しない $\Rightarrow$ No

a_1	3
a_2	7
a_3	2
a_4	4
a_5	3
a_6	2
a_7	1
t	9

これは Yes 入力：$X = \{2, 3\}$ が解

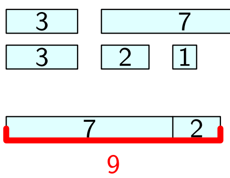
- $\sum_{i \in X} a_i = a_2 + a_3 = 7 + 2 = 9 = t$

部分和問題

部分和問題 (subset sum problem)

- 入力：n 個の自然数の集合 $A = \{a_1, a_2, \ldots, a_n\} \subseteq \mathbb{N}$, 自然数 $t \in \mathbb{N}$
- 出力：Yes か No
- 条件：$\sum_{i \in X} a_i = t$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在する $\Rightarrow$ Yes
  $\sum_{i \in X} a_i = t$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在しない $\Rightarrow$ No

a_1	3
a_2	7
a_3	2
a_4	4
a_5	3
a_6	2
a_7	1
t	9



部分和問題の NP 完全性

定理 (Karp '72)

部分和問題は弱 NP 完全

これを証明するためには，次を証明すればよい

- 入力数値が 2 進符号化されるとき，部分和問題が NP に所属すること
- 入力数値が 2 進符号化された 2 分割問題が
  入力数値が 2 進符号化された部分和問題に
  多項式時間多対一帰着可能であること

部分和問題の NP 完全性：NP に所属すること

部分和問題 (subset sum problem)

- 入力：n 個の自然数の集合 $A = \{a_1, a_2, \ldots, a_n\} \subseteq \mathbb{N}$, 自然数 $t \in \mathbb{N}$
- 出力：Yes か No
- 条件：$\sum_{i \in X} a_i = t$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在する $\Rightarrow$ Yes
  $\sum_{i \in X} a_i = t$ となる $X \subseteq \{1, 2, \ldots, n\}$ が存在しない $\Rightarrow$ No

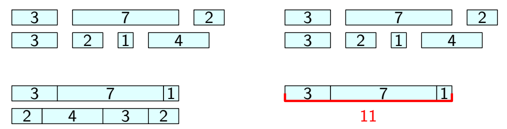
Yes 入力の証拠：$\sum_{i \in X} a_i = t$ となる集合 $X \subseteq \{1, 2, \ldots, n\}$

多項式時間検証アルゴリズム

1. $\sum_{i \in X} a_i = t$ となるか，確認

2 進法の加算は多項式時間で実行できるので，
この検証アルゴリズムは多項式時間で実行できる

部分和問題の NP 完全性：多対一多項式時間帰着



2 分割問題の入力 $a'_1, \ldots, a'_n$ から部分和問題の入力 $a_1, \ldots, a_n, t$ を構成

- すべての $i \in \{1, 2, \ldots, n\}$ に対して，$a_i = a'_i$
- $t = \frac{1}{2} \sum_{i=1}^{n} a_i$ (この構成は多項式時間で可能)

このとき，次の 2 つは同値

- $X \subseteq \{1, 2, \ldots, n\}$ が $\sum_{i \in X} a'_i = \sum_{i \notin X} a'_i$ を満たす
- $X \subseteq \{1, 2, \ldots, n\}$ が $\sum_{i \in X} a_i = t$ を満たす □

ナップサック問題 (最適化問題)

ナップサック問題：価値を最大にするように，モノを詰めたい

ナップサック問題 (knapsack problem)

- 入力：n 個のアイテム $\{1, 2, \ldots, n\}$
  各アイテム i の重さ $w_i \in \mathbb{N}$ と価値 $p_i \in \mathbb{N}$
  ナップサックの容量 $C \in \mathbb{N}$
- 出力：$\sum_{i \in X} w_i \leq C$ を満たす集合 $X \subseteq \{1, 2, \ldots, n\}$
- 評価：$\sum_{i \in X} p_i$ の最大化

アイテム	1	2	3	4	5
重さ w_i	4	2	5	3	3
価値 p_i	3	1	4	3	2

- 容量 $C = 8$

最適解は $X = \{3, 4\}$

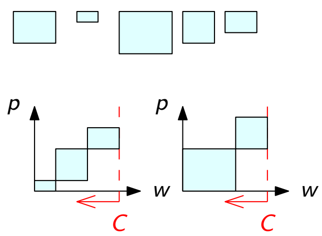
- $w_3 + w_4 = 8 \leq C$
- $p_3 + p_4 = 7$

ナップサック問題 (最適化問題)

ナップサック問題：価値を最大にするように，モノを詰めたい

アイテム	1	2	3	4	5
重さ w_i	4	2	5	3	3
価値 p_i	3	1	4	3	2

- 容量 $C = 8$



ナップサック問題 (判定問題版)

ナップサック問題 (判定問題版)

- 入力：n 個のアイテム $\{1, 2, \dots, n\}$
  各アイテム i の重さ $w_i \in \mathbb{N}$ と価値 $p_i \in \mathbb{N}$
  ナップサックの容量 $C \in \mathbb{N}$，自然数 $k \in \mathbb{N}$
- 出力：Yes または No
- 条件：$\sum_{i \in X} w_i \le C$ かつ $\sum_{i \in X} p_i \ge k$ を満たす集合 $X \subseteq \{1, 2, \dots, n\}$ が
  存在する ⇒ Yes
  そうでない ⇒ No

アイテム	1	2	3	4	5
重さ w_i	4	2	5	3	3
価値 p_i	3	1	4	3	2

- 容量 $C = 8$
- $k = 5$

$X = \{1, 4\}$ とすると

- $w_1 + w_4 = 7 \le C$
- $p_1 + p_4 = 6 \ge k$

つまり，これは Yes 入力

ナップサック問題の NP 完全性

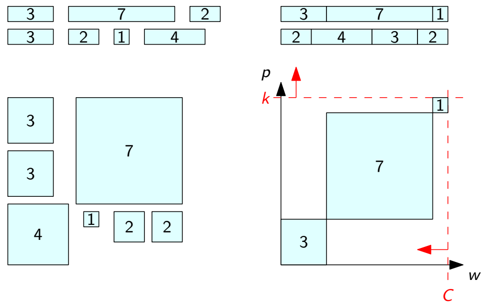
定理 (Karp '72)

ナップサック問題 (判定問題版) は弱 NP 完全

これを証明するためには，次を証明すればよい

- 入力数値が 2 進符号化されるとき，
  ナップサック問題 (判定問題版) が NP に所属すること
  (これは 部分和問題と同様に 証明できる)
- 入力数値が 2 進符号化された 2 分割問題が
  入力数値が 2 進符号化されたナップサック問題 (判定問題版) に
  多項式時間多対一帰着可能であること

ナップサック問題の NP 完全性：多対一多項式時間帰着 (1)



ナップサック問題の NP 完全性：多対一多項式時間帰着 (2)

2 分割問題の入力 $\{a'_i\}$ から
ナップサック問題の入力 $\{w_i\}, \{p_i\}, C, k$ を構成

- すべての $i \in \{1, 2, \dots, n\}$ に対して，$w_i = a'_i$，$p_i = a'_i$
- $C = k = \frac{1}{2} \sum_{i=1}^{n} a'_i$　　(この構成は多項式時間で可能)

このとき，次の 2 つは同値

- $X \subseteq \{1, 2, \dots, n\}$ が $\sum_{i \in X} a'_i = \sum_{i \notin X} a'_i$ を満たす
- $X \subseteq \{1, 2, \dots, n\}$ が $\sum_{i \in X} w_i \le C$ と $\sum_{i \in X} p_i \ge k$ を満たす　□

2 分割問題に似た問題：補足

ここで証明したこと

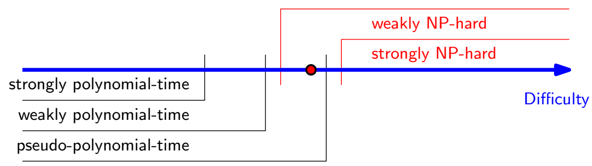
次の 2 つの問題は弱 NP 完全

- 部分和問題
- ナップサック問題 (判定問題版)

事実 (Bellman '56, Dantzig '57)

この 2 つの問題を解く擬多項式時間アルゴリズムが存在する

2 分割問題と同じように，アルゴリズムを設計できる



目次

スケジューリング問題

スケジューリング問題には，様々な設定，様々な変種がある

ここで扱うものは，その中でも，最も基本的なもの

共通の設定

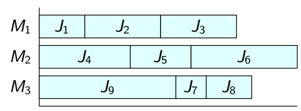
- 処理すべきジョブが複数ある：$J_1, J_2, \dots, J_n$
- 処理をする機械が (複数) ある：$M_1, M_2, \dots, M_m$
  - 各機械の性能は同一 (同一機械スケジューリング)
- 各ジョブ J_i の処理時間 $= p_i$
- 各ジョブを 1 つの機械で処理する必要があり，1 回処理すれば十分
- 割り込みなし (no preemption)
  - ジョブの処理を一旦開始したら，中断できない

このとき，スケジューリングとは，
各機械で，どのジョブをどの順で処理するか決めること

スケジューリング問題：イメージ

このような図をガント・チャート (Gantt chart) と呼ぶことがある

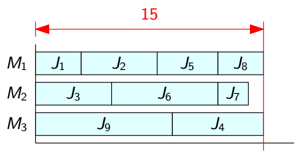
$p_1 = 3,\quad p_2 = 5,\quad p_3 = 5,\quad p_4 = 6,\quad p_5 = 4,$
$p_6 = 7,\quad p_7 = 2,\quad p_8 = 3,\quad p_9 = 9$



スケジューリング問題：最終完了時刻最小化

よくある目的：最小完了時刻 (makespan) の最小化

$p_1 = 3,\quad p_2 = 5,\quad p_3 = 5,\quad p_4 = 6,\quad p_5 = 4,$
$p_6 = 7,\quad p_7 = 2,\quad p_8 = 3,\quad p_9 = 9$



最終完了時刻最小化問題

m は あらかじめ決められた定数

同一 m 機械スケジューリングにおける最終完了時刻最小化問題

- 入力：m 個の同一機械，n 個のジョブ $J_1, \dots, J_n$
  各ジョブ J_i の処理時間 $p_i \in \mathbb{N}$
- 出力：m 個の機械へのスケジューリング (割り込みなし)
- 評価：最終完了時刻の最小化

同一 m 機械スケジューリングにおける最終完了時刻最小化問題 (判定問題版)

- 入力：m 個の同一機械，n 個のジョブ $J_1, \dots, J_n$
  各ジョブ J_i の処理時間 $p_i \in \mathbb{N}$，期限 $T \in \mathbb{N}$
- 出力：Yes または No
- 条件：最終完了時刻が T 以下のスケジューリングがある ⇒ Yes
  最終完了時刻が T 以下のスケジューリングがない ⇒ No

最終完了時刻最小化問題の NP 完全性

m は あらかじめ決められた定数

定理 (Garey, Johnson '79)

$m \geq 2$ のとき，同一 m 機械スケジューリングにおける最終完了時刻最小化問題 (判定問題版) は弱 NP 完全

これを証明するためには，次を証明すればよい

- 入力数値が 2 進符号化されるとき，
  最終完了時刻最小化問題 (判定問題版) が NP に所属すること
  (これは 部分和問題と同様に 証明できる)
- 入力数値が 2 進符号化された 2 分割問題が
  入力数値が 2 進符号化された最終完了時刻最小化問題 (判定問題版)
  に 多項式時間多対一帰着可能であること

最終完了時刻最小化問題の NP 完全性：多項式時間多対一帰着 (1)

最終完了時刻最小化問題の NP 完全性：多項式時間多対一帰着 (2)

2 分割問題の入力から最大完了時刻最小化問題の入力を構成

- ジョブの数は $n + m - 2$ で，
  すべての $i \in \{1, 2, \dots, n\}$ に対して，$p_i = a'_i$，
  すべての $i \in \{n+1, \dots, n+m-2\}$ に対して，$p_{n+1} = \frac{1}{2}\sum_{i=1}^{n} a'_i$
- $T = \frac{1}{2}\sum_{i=1}^{n} a'_i$　(この構成は多項式時間で可能)

このとき，次の 2 つは同値

- $X \subseteq \{1, 2, \dots, n\}$ が $\sum_{i \in X} a'_i = \sum_{i \notin X} a'_i$ を満たす
- M_1 に $\{J_i \mid i \in X\}$ を，M_2 に $\{J_i \mid i \notin X\}$ を，
  M_i に $\{J_{n+i-2}\}$ $(i \in \{3, \dots, m\})$ を割り当てると，
  最大完了時刻が T となる □

最終完了時刻最小化問題：まとめ

m は あらかじめ決められた定数

定理 (Garey, Johnson '79)

$m \geq 2$ のとき，同一 m 機械スケジューリングにおける最終完了時刻最小化問題 (判定問題版) は弱 NP 完全

事実 (Sahni '76)

$m \geq 2$ のとき，同一 m 機械スケジューリングにおける最終完了時刻最小化問題を解く擬多項式時間アルゴリズムが存在する

- 最悪時実行時間 $= O\left(nT^{m-1}\right)$　(これも動的計画法)

$\boxed{\text{注}}$：この最悪時実行時間は，m が定数であるから擬多項式

目次

今日のまとめ と 次回の予告

今日の目標

- 数値が関わる問題に関する基礎概念を理解する
  - 数値の入力法：2 進法，1 進法
  - 強多項式時間，弱多項式時間，擬多項式時間
  - 強 NP 完全，弱 NP 完全
- 数値が関わる問題の中で最も基礎的な NP 完全問題を理解する
  - 2 分割問題
- 数値が関わる問題について，NP 完全性の証明を行う
  - 部分和問題，ナップサック問題，スケジューリング問題

次回の予告

- 数値が関わる強 NP 完全問題を扱う
  - 基礎となる問題：3 分割問題
  - スケジューリング問題
  - 3 分割問題を用いた帰着

- R. Bellman, Notes on the theory of dynamic programming IV – maximization over discrete sets. Naval Research Logistics Quarterly 3 (2956) pp. 67–70.
- G. B. Dantzig, Disrete variable extremum problems. Operations Research 5 (1957) pp. 266–277.
- M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., 1979.
- R. M. Karp, Reducibility among combinatorial problems. In: R. E. Miller, J. W. Thatcher, J. D. Bohlinger (eds.), Complexity of Computer Computations. New York, Plenum (1972) pp. 85–103.
- S. K. Sahni, Algorithms for scheduling independent tasks. Journal of the ACM 23 (1976) pp. 114–127.