

- \* 休み (大学院入学式) (10/1)
- 1 アルゴリズム的問題解決と計算複雑性 (10/8)
- 2 速習 P vs NP 問題 (10/15)
- \* 休み (祝日) (10/22)
- 3 充足可能性問題とその変種 (10/29)
- 4 グラフに関する問題 (1) : 部分集合の選択 (11/5)
- 5 グラフに関する問題 (2) : 経路の選択 (11/12)
- 6 集合族に関する問題 (1) : グラフとの関連 (11/19)
- 7 集合族に関する問題 (2) : 発展 (11/26)

### スケジュール 後半 (予定)

### 今日の目標

- 8 数値が関わる問題 (1) : 2 分割問題 (12/3)
- 9 数値が関わる問題 (2) : 3 分割問題 (12/10)
- 10 平面性が関わる問題 (12/17)
- \* 冬期休業 (12/24, 31)
- 11 計算幾何学に関する問題 (1/7)
- 12 文字列に関する問題 (1/14)
- 13 アルゴリズム的問題解決 : 再考 (1/21)
- 14 予備 (1/28)
- \* 休講 (2/4)
- \* 祝日のため休み (2/11)

注意 : 予定の変更もありうる

### 今日の目標

- 文字列に関する問題の NP 困難性を扱う
- ▶ 最長共通部分列問題
  - ▶ 文字列のクラスタリング問題

### 文字列はどこに現れるか?

### 目次

#### 文字列はどこに現れるか?

- どこにでも現れる。特に
- ▶ 分子生物学 (DNA の一次構造) ( $\Sigma = \{A, G, C, T\}$ )
  - ▶ 自然言語処理
  - ▶ オートマトン理論と形式言語学 (形式文法)
  - ▶ 符号理論, データ圧縮 ( $\Sigma = \{0, 1\}$ )

文字列はとても基本的な対象なので、  
数学的性質やアルゴリズムが活発に研究されている  
→ 文字列学 (stringology)

- 1 文字列の基礎用語
- 2 最長共通部分列問題
- 3 文字列間の距離とクラスタリング問題
- 4 今日のまとめ と 次回の予告

### 文字列とは?

### 文字列の連結と反復

#### 文字列 : 直感的な定義

- ▶ **文字列** (string) は文字を並べたもの (語 (word) と呼ぶこともある)
- ▶ **文字** (character, letter) はある有限集合の要素
- ▶ **アルファベット** (alphabet) は使う文字全体の集合

アルファベットが  $\Sigma = \{a, b, c, d\}$  であるとき

- ▶  $abcbaacd$  は  $\Sigma$  上の文字列 (長さは 7,  $|abcbaacd| = 7$ )
- ▶  $ccddabc$  は  $\Sigma$  上の文字列
- ▶  $abcdeab$  は  $\Sigma$  上の文字列ではない ( $\because e \notin \Sigma$ )

文字列  $s, t \in \Sigma^*$

#### 定義 : 文字列の連結 (結合, concatenation)

- ▶  $st =$  文字列  $s$  と文字列  $t$  をこの順に連結してできる文字列

例 :  $s = abba, t = cdcd$  のとき

- ▶  $st = abbaacdcd$
- ▶  $ts = cdcdabba$

#### 定義 : 文字列の反復 (repetition)

- ▶  $s^n =$  文字列  $s$  を  $n$  回反復してできる文字列

例 :  $s = abba$  のとき

- ▶  $s^3 = abbaabbaabba$
- ▶  $s^2 = abbaabba$
- ▶  $s^1 = abba$
- ▶  $s^0 = \varepsilon$

#### 記法 : 文字列の集合

- ▶  $\Sigma^n = \Sigma$  上の文字列で、長さ  $n$  のもの全体の集合
- ▶  $\Sigma^* = \Sigma$  上の文字列全体の集合 (長さは問わない)

$\varepsilon =$  長さ 0 の文字列 (空文字列)

長さ  $n$  の文字列  $s$

- ▶  $s[i] = s$  の先頭から  $i$  番目の文字 ( $i \in \{1, 2, \dots, n\}$ )

定義:  $s$  の部分文字列 (substring)

ある  $i, j \in \{1, 2, \dots, n\}$  ( $i \leq j$ ) に対して, 次のように書ける文字列

$$s[i]s[i+1]\dots s[j]$$

定義:  $s$  の部分列 (subsequence)

ある  $i_1, i_2, \dots, i_j \in \{1, 2, \dots, n\}$  ( $i_1 < i_2 < \dots < i_j$ ) に対して, 次のように書ける文字列

$$s[i_1]s[i_2]\dots s[i_j]$$

例:  $s = abbaabbaabba$  のとき

- ▶  $bbaab$  は  $s$  の部分文字列であり,  $s$  の部分列である
- ▶  $aaaab$  は  $s$  の部分文字列ではないが,  $s$  の部分列である

- ① 文字列の基礎用語
- ② 最長共通部分列問題
- ③ 文字列間の距離とクラスタリング問題
- ④ 今日のまとめ と 次回の予告

最長共通部分列問題 (longest common subsequence problem)

最長共通部分列問題 (最適化問題版)

- ▶ 入力: 整数  $n \geq 1$ , アルファベット  $\Sigma$ , 文字列  $s_1, s_2, \dots, s_n \in \Sigma^*$
- ▶ 出力:  $s_1, s_2, \dots, s_n$  の共通部分列  $t$
- ▶ 評価:  $t$  の長さの最大化

$s_1 = abcabcabcabc$

$s_2 = aabaabbcabab$

$s_3 = aaaabbbbcccc$

$\rightsquigarrow t = aaabbc$

最長共通部分列問題 (判定問題版)

最長共通部分列問題 (判定問題版)

- ▶ 入力: 整数  $n \geq 1$ , アルファベット  $\Sigma$ , 文字列  $s_1, s_2, \dots, s_n \in \Sigma^*$   
整数  $d \geq 0$
- ▶ 出力: Yes または No
- ▶ 条件:  $s_1, s_2, \dots, s_n$  の共通部分列で長さ  $d$  のものがある  $\Rightarrow$  Yes  
そうでない  $\Rightarrow$  No

$s_1 = abcabcabcabc$

$s_2 = aabaabbcabab$

$s_3 = aaaabbbbcccc$

$\rightsquigarrow t = aaabbc$  (長さ 6)

最長共通部分列問題の NP 完全性

定理 (Maier '78)

最長共通部分列問題 (判定問題版) は NP 完全

これを証明するためには, 次のことを証明すればよい

- ▶ 最長共通部分列問題 (判定問題版) が NP に所属すること
- ▶ 最小共通部分列問題 (判定問題版) に  
独立集合問題が多項式時間多対一帰着可能であること

最長共通部分列問題の NP 完全性: NP に所属すること

Yes 入力の証拠: 長さ  $d$  の共通部分列  $t$

多項式時間検証アルゴリズム

- 1  $t$  の長さが  $d$  であることを確認
- 2  $t$  が  $s_1, s_2, \dots, s_n$  の部分列であること確認

(ただし, 2つの文字の比較は多項式時間でできるとする)

$s_1 = abcabcabcabc$

$s_2 = aabaabbcabab$

$s_3 = aaaabbbbcccc$

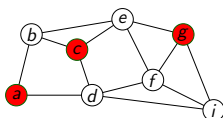
$\rightsquigarrow t = aaabbc$

復習: 独立集合問題

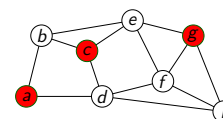
独立集合問題

- ▶ 入力: 無向グラフ  $G$ , 自然数  $k \in \mathbb{N}$
- ▶ 出力: Yes か No
- ▶ 条件:  $G$  が要素数  $k$  以上の独立集合を持つ  $\Rightarrow$  Yes  
 $G$  が要素数  $k$  以上の独立集合を持たない  $\Rightarrow$  No

無向グラフ  $G$  の独立集合とは,  $G$  の頂点部分集合  $S$  で  $S$  のどの2頂点間にも辺が存在しないもの



最長共通部分列問題の NP 完全性: 多対一多項式時間帰着 (1)



$\Sigma =$  頂点集合  $\{a, b, c, d, e, f, g, i\}$

- |                                    |                                       |
|------------------------------------|---------------------------------------|
| $s_0: a b c d e f g i$             | $s_7: a b c e f g i a b c d e g i$    |
| $s_1: b c d e f g i a c d e f g i$ | $s_8: a b c e f g i a b c d e f g$    |
| $s_2: b c d e f g i a b c e f g i$ | $s_9: a b c d f g i a b c d e g i$    |
| $s_3: a c d e f g i a b d e f g i$ | $s_{10}: a b c d f g i a b c d e f i$ |
| $s_4: a c d e f g i a b c d f g i$ | $s_{11}: a b c d e g i a b c d e f i$ |
| $s_5: a b d e f g i a b c e f g i$ | $s_{12}: a b c d e g i a b c d e f g$ |
| $s_6: a b d e f g i a b c d f g i$ | $s_{13}: a b c d e f i a b c d e f g$ |

無向グラフ  $G = (V, E)$  から、次のように文字列を構成する

- ▶  $V = \{v_1, v_2, \dots, v_n\}, E = \{e_1, e_2, \dots, e_m\}$  とする
- ▶  $\Sigma = V$
- ▶  $s_0 = v_1 v_2 v_3 \dots v_n$  (長さ  $n$ )
- ▶ 各  $i \in \{1, 2, \dots, m\}$  に対して,  $e_i = \{v_j, v_k\} (j < k)$  であるとする  
 $s_i = v_1 v_2 v_3 \dots v_{j-1} v_{j+1} \dots v_n v_1 v_2 v_3 \dots v_{k-1} v_{k+1} \dots v_n$  (長さ  $2n - 2$ )

この構成は多項式時間で行なえて,

$G$  が要素数  $d$  の独立集合を持つ  $\Leftrightarrow s_0, s_1, \dots, s_m$  が長さ  $d$  の共通部分列を持つ

特に

$\{v_{i_1}, v_{i_2}, \dots, v_{i_d}\}$  が  $G$  の独立集合である ( $i_1 < i_2 < \dots < i_d$ )  $\Leftrightarrow$  文字列  $v_{i_1} v_{i_2} \dots v_{i_d}$  が  $s_0, s_1, \dots, s_m$  の共通部分列である



## 目次

- 1 文字列の基礎用語
- 2 最長共通部分列問題
- 3 文字列間の距離とクラスタリング問題
- 4 今日のまとめ と 次回の予告

## 2つの文字列の間の距離 (1) : Hamming 距離

Hamming 距離は、同じ長さの文字列どうしの距離

定義 : Hamming 距離

長さ  $n$  の 2 つの文字列  $s, t \in \Sigma^n$  の Hamming 距離とは

$$HD(s, t) = |\{i \in \{1, 2, \dots, n\} \mid s[i] \neq t[i]\}|$$

つまり,  $s, t$  で異なる文字を持つ添え字の総数

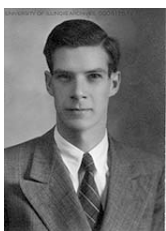
例 :  $HD(drunk, blank) = 3$

d	r	u	n	k
b	l	a	n	k

事実

$HD(s, t)$  は  $O(n)$  時間で計算できる (ただし, 2 つの文字の比較は  $O(1)$  でできるものとする)

## Richard Hamming と Vladimir Levenshtein



リチャード・ハミング (1915-1998)



ウラジミール・レーベンシュタイン (1935-2007)

<https://archon.library.illinois.edu/?p=digitallibrary/digitalcontent&id=11804>  
[https://www.keldysh.ru/departments/dpt\\_10/lev.html](https://www.keldysh.ru/departments/dpt_10/lev.html)

定理

(Maier '78)

最長共通部分列問題 (判定問題版) は NP 完全

実は, 次のような制限があっても, 最長共通部分列問題は NP 完全

- ▶  $|\Sigma| = 2$  であっても (Maier '78)
- ▶  $s_1, s_2, \dots, s_n$  が同じ偶数であっても (de la Higuera, Casacuberta '00)
- ▶  $|\Sigma| = 2$  であり, かつ,  $s_1, s_2, \dots, s_n$  が同じ偶数であっても (Nicolas, Rivals '05)

## 文字列に対する直感 1

「 $\Sigma$  が可変である場合」が難しい問題は, 「 $\Sigma$  が固定である場合」も難しいことが多い

## 文字列の間の距離

復習 : 距離とは ?

$V$  上の距離 (metric) とは, 次の満たす関数  $d: V^2 \rightarrow \mathbb{R}$  のこと

- ▶ 任意の  $u, v \in V$  に対して,  $d(u, v) \geq 0$
- ▶ 任意の  $u, v \in V$  に対して,  $d(u, v) = 0 \Leftrightarrow u = v$
- ▶ 任意の  $u, v \in V$  に対して,  $d(u, v) = d(v, u)$
- ▶ 任意の  $u, v, w \in V$  に対して,  $d(u, v) \leq d(u, w) + d(w, v)$

Q: どうして 文字列の間の距離 を考えたい ?

A: 文字列の集合に対して, クラスタリングなどを行いたいから  
 $\rightsquigarrow$  データマイニング

文字列に対する距離はたくさん考えられている

代表的なもの

- 1 Hamming 距離
- 2 Levenshtein 距離 (編集距離)

## 2つの文字列の間の距離 (2) : Levenshtein 距離 (編集距離)

Levenshtein 距離は, 文字列の長さが異なっていても定義できる

定義 : Levenshtein 距離 (編集距離)

2 つの文字列  $s, t \in \Sigma^*$  の Levenshtein 距離  $LD(s, t)$  とは  $s$  から  $t$  を得るために行う文字の置換, 挿入, 削除の回数の最小値

例 :  $LD(kitten, sitting) = 3$

- 1 kitten  $\rightarrow$  sitten (置換)
- 2 sitten  $\rightarrow$  sittin (置換)
- 3 sittin  $\rightarrow$  sitting (挿入)

事実

(Wagner, Fisher '74)

$LD(s, t)$  は  $O(|s||t|)$  時間で計算できる (ただし, 2 つの文字の比較は  $O(1)$  でできるものとする)

補足 : Hamming 距離は「置換」のみを行う場合と考えることができる

## 最近接文字列問題 (closest string problem)

「Hamming 距離におけるクラスタ中心を求める」という問題を考える (Hamming 距離に関する 1-センター問題)

最近接文字列問題 (判定問題版)

- ▶ 入力 : 整数  $n, \ell \geq 1$ , アルファベット  $\Sigma$ , 文字列  $s_1, s_2, \dots, s_n \in \Sigma^\ell$  整数  $d \geq 0$
- ▶ 出力 : Yes または No
- ▶ 条件 : すべての  $i$  で  $HD(s_i, t) \leq d$  となる文字列  $t \in \Sigma^\ell$  がある  $\Rightarrow$  Yes そうでない  $\Rightarrow$  No

$s_1 = akasaka \quad HD(s_1, t) = 2$

$s_2 = asakusa \quad HD(s_2, t) = 2$

$s_3 = isasaka \quad HD(s_3, t) = 2$

$\rightsquigarrow t = asasuka$

「Levenshtein 距離におけるクラスタ中心を求める」という問題を考える (Levenshtein 距離に関する 1-センター問題)

中心文字列問題 (判定問題版)

- ▶ 入力: 整数 n ≥ 1, アルファベット Σ, 文字列 s1, s2, ..., sn ∈ Σ\*
整数 d ≥ 0
▶ 出力: Yes または No
▶ 条件: すべての i で LD(si, t) ≤ d となる文字列 t ∈ Σ\* がある ⇒ Yes
そうでない ⇒ No

s1 = niger LD(s1, t) = 2
s2 = nigeria LD(s2, t) = 2
s3 = algeria LD(s3, t) = 2
⇨ t = aigeri

中心文字列問題の NP 完全性: 帰着

s1, ..., sn の長さは 2l であるとする

s0: ε t: 0 0 1 1 0
s1: 0 0 0 0 1 1 1 1 0 1
s2: 0 0 1 1 1 1 1 1 0 0
s3: 0 1 0 1 0 1 0 1 0 1
s4: 0 0 1 1 0 1 1 1 1 1
s5: 1 1 0 0 1 1 1 0 0 1

s1, ..., sn が長さ l の
共通部分列を持つ ⇔ LD(si, t) ≤ l となる
文字列 t が存在する

定理 (Frances, Litman '97)

最近接文字列問題 (判定問題版) は NP 完全

▶ |Σ| = 2 であっても NP 完全 (Frances, Litman '97)

3-SAT を帰着して, 証明する

定理 (de la Higuera, Casacuberta '00)

中心文字列問題 (判定問題版) は NP 完全

▶ |Σ| = 2 であっても NP 完全 (Nicolas, Rivals '05)

最長共通部分列問題 (入力文字列の長さが同じ偶数の場合) を帰着して,
証明する

最近接文字列問題の NP 完全性: 帰着

f = (x1 ∨ x3 ∨ x4) ∧ (x1 ∨ x2 ∨ x4) ∧ (x2 ∨ x3 ∨ x4)

t: 0 0 1 1 0 0 1 1 0 0 t: 0 0 1 1 0 0 1 1 0 0 t: 0 0 1 1 0 0 1 1 0 0
0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0 0
1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 0 1 0 0
0 1 1 0 0 1 0 1 0 1 1 0 0 1 1 0 1 0 1 1 1 1 1 0 0
0 1 0 1 1 0 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 1 0 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 1 0 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

f が充足可能 ⇔ 構成した各文字列 s に対して
HD(s, t) ≤ n + 1 となる文字列 t が存在
(n は f の変数の数)

最近接文字列問題 / 中心文字列問題の NP 完全性 (再掲)

最近接文字列問題 / 中心文字列問題は文字列に対する 1-センター問題

定理 (Frances, Litman '97)

最近接文字列問題 (判定問題版) は NP 完全

▶ |Σ| = 2 であっても NP 完全 (Frances, Litman '97)

3-SAT を帰着して, 証明する

定理 (de la Higuera, Casacuberta '00)

中心文字列問題 (判定問題版) は NP 完全

▶ |Σ| = 2 であっても NP 完全 (Nicolas, Rivals '05)

最長共通部分列問題 (入力文字列の長さが同じ偶数の場合) を帰着して,
証明する

Hamming 距離 vs Levenshtein 距離

文字列に対する直感 2

「Hamming 距離に対する問題」は「Levenshtein 距離に対する問題」と
同じ程度以上に難しい

難しさを不等号で表すと

Levenshtein 距離に対する問題 ≥ Hamming 距離に対する問題

例:

Table with 3 columns: Problem, Levenshtein 距離, Hamming 距離. Rows include 1-センター問題, 1-メディアン問題.

s1 = akasaka HD(s1, t) = 1
s2 = asakusa HD(s2, t) = 3
s3 = isasaka HD(s3, t) = 1
⇨ t = asasaka 和 = 4

目次

- 1 文字列の基礎用語
2 最長共通部分列問題
3 文字列間の距離とクラスタリング問題
4 今日のまとめ と 次回予告

今日のまとめ と 次回の内容

今日の目標

文字列に関する問題の NP 困難性を扱う

- ▶ 最長共通部分列問題
▶ 文字列のクラスタリング問題

次回予告

全体のまとめ と 最近の話題

- ▶ 「NP 完全問題」であると分かったら, どうすればよいのか?
▶ 「多項式時間で解ける」と分かったら, それでよいのか?

- ▶ C. de la Higuera, F. Casacuberta, Topology of strings: median string is NP-complete *Theoretical Computer Science* 230 (2000) pp. 39–48.
- ▶ M. Frances, A. Litman, On covering problems of codes. *Theory of Computing Systems* 30 (1997) pp. 113–119.
- ▶ D. Maier, The complexity of some problems on subsequences and supersequences. *Journal of the Association for Computing Machinery* 25 (1978) pp. 322–336.
- ▶ F. Nicolas, E. Rivals, Hardness results for the center and median string problems under the weighted and unweighted distances. *Journal of Discrete Algorithms* 3 (2005) pp. 390–415.
- ▶ R. Wagner, M. Fisher, The string-to-string correction problem. *Journal of the Association for Computing Machinery* 21 (1974) pp. 168–178.