



מבני נתונים ואלגוריתמים

התאמת מחרוזות

התאמת מחרוזות-מושגים

- **טקסט T** (מערך של תווים) באורך n – $T[0, \dots, n-1]$ ו**תבנית P** באורך m – $P[0, \dots, m-1]$ כך ש- $m \leq n$.
- התווים של P ו- T נלקחים מאלפבית סופי Σ . לדוגמא: $\Sigma = \{0, 1\}$, $\{a, b, \dots, z\}$.
- Σ^* - קבוצת כל המחרוזות באורך סופי שניתן להרכיב מ- Σ .
- ε - המחרוזת הריקה. מתקיים ש- $\varepsilon \in \Sigma$

התאמת מחרוזות – תיאור הבעיה

- רוצים למצוא את כל המופעים של P ב-T.
- רוצים למצוא את כל האינדקסים (=היסטים) ב-T כך ש-

$$T[i, \dots, i + m - 1] = P[0, \dots, m - 1]$$

לדוגמא:

T טקסט

a	a	b	a	b	b	b	c	a	b	b	a	c
---	---	---	---	---	---	---	---	---	---	---	---	---

P תבנית

a	b	b
---	---	---

S=3 →

התאמת מחרוזות – עוד מושגים

- תהי X מחרוזת. נסמן ב- X_i את המחרוזת שמכילה את התווים מ-0 עד i ב- X ואורכה i . כלומר
$$X_i = X[0, \dots, i-1]$$
- **רישא:** x רישא של y אם קיים z כך ש- $xz = y$ (=שרשור של x עם z).
- **סיפא:** x סיפא של y אם קיים z כך ש- $zx = y$.

אלגוריתמים

- האלגוריתם הנאיבי $O(nm)$
- אלגוריתם רבין - קארפ $O(n+m)$
- אלגוריתם KMP $O(n+m)$
- אלגוריתם BM $O(n/m)$
- עצי סיפא

רבין קארפ

- נסתכל על הטקסט כמספרים ולא כאותיות.
- נרצה לייצג כל מחרוזת כמספר בעזרת פונקציית hash
- **הרעיון** – נשתמש בפונקציית hash. נחשב את ערך ה-hash של התבנית, ונשווה אותו לערך ה-hash של כל תתי-המחרוזות בטקסט.

רבין קארפ - בעיה

נחשב את $H(P)$ - לוקח $O(m)$ כאשר m אורך התבנית.
עוברים על כל תתי המחרוזות t ב- t מחשבים את $H(t)$ ומשווים ל- $H(P)$
נקבל $O(nm)$!

ROLLING HASH

זוהי פונקציית Hash מיוחדת המחשבת את ערך ה-Hash עבור תת-המחרוזת הראשונה ב- t בזמן $O(m)$ (כאורך התבנית) ואז מחשבת את תת-המחרוזת הבאה מתוך תת-המחרוזת הראשונה ב- $O(1)$ וכך הלאה.

נקבל $O(m+n)$

ROLLING HASH

בהרצאה ראיתם הצעה לפונקציית hash :

$$f(V) = \sum_{i=0}^{L-1} V[i] * A^i \% B$$

כאשר A הוא מספר ראשוני גדול.

תרגיל: חשבו את

$$f(\textit{banana})$$

ROLLING HASH

בהרצאה ראיתם הצעה לפונקציית hash :

$$f(V) = \sum_{i=0}^{L-1} V[i] * A^i \% B$$

כאשר A הוא מספר ראשוני גדול.

תרגיל: חשבו את

$$f(banana)$$

פתרון:

$$f(banana) = 2 * A^0 + 1 * A^1 + 14 * A^2 + 1 * A^3 + 14 * A^4 + 1 * A^5$$

ROLLING HASH

ראינו ש:

$$f(\textit{banana}) = 2 * A^0 + 1 * A^1 + 14 * A^2 + 1 * A^3 + 14 * A^4 + 1 * A^5$$

מה יקרה אם נרצה לחשב את $f(\textit{ananaa})$

$$f(\textit{ananaa}) = \frac{f(\textit{banana}) - f(\textit{b})}{A} + f(\textit{a}) * A^5$$

פסאדו קוד

Init

$i=0$

$Z=f(w)$

$Q=f(T(0:i+L-1))$

While $i < K-L$:

 If $Z==Q$:

 compare and if True save l

 else:

$i++$

 compute $f(T(i:i+L-1))$ with rolling hash

l טקסט בארוך k ו w מילה בארוך L

תרגיל:

נתונות 2 מחרוזות X, Y באורך n . X אנגרמה של Y אם קיימת תמורה $p \in S_{\{0,1,\dots,n-1\}}$ כך שמתקיים

$$Y = X[p(0)] X[p(1)] \dots X[p(n-1)]$$

כתבו אלגוריתם שמקבל T ו- P ומוצא את כל תתי המחרוזות של T שהן אנגרמות של P .

פתרון

פיתרון:

נריץ RB עם Rolling hash שמחשב את ערך ה-Hash על המחרוזת ללא תלות בסדר האותיות ואז אם x ו- y הן אנגרמות אז $RH(x) = RH(y)$.

לדוגמא – לכל אות $a \in \Sigma$ נבחר מספר גדול N_a (כדי שסכום 2 מספרים יהיה ייחודי) ונגדיר:

$$RH(s) = \sum_{i=0}^{len(s)-1} N_{s[i]}$$



KMP

המטרה: להשתמש בידע שצברנו על התבנית P כדי לחסוך בבדיקות.
רעיון: נחשב את פונקציית הרישא של התבנית!

שאלה: מה זה פונקציית הרישא?

על מנת להימנע מהשוואות מיותרות, נשתמש במידע שיש בתבנית P עצמה. מחשבים את פונקציית Π – פונקציית הרישא עבור תבנית P באורך m.

$\Pi(i) = j$ – האינדקס המקסימלי j ($0 \leq j < i$) המקיים ש- P_j היא סיפא של P_i (לכל $0 \leq i < m$).

דוגמא:

	0	1	2	3	4	5	
P = a b a b c	Π	-1	0	0	1	2	0

$\Pi(1)$ – צריך למצוא את הרישא הכי ארוכה של $P_0 = \epsilon$ שהיא סיפא של $P_1 = a$. אין - לכן $\Pi(1) = 0$.

$\Pi(2)$ – צריך למצוא את הרישא הכי ארוכה של $P_1 = a$ שהיא סיפא של $P_2 = ab$. אין - לכן $\Pi(2) = 0$.

$\Pi(3)$ – צריך למצוא את הרישא הכי ארוכה של $P_2 = ab$ שהיא סיפא של $P_3 = aba$, לכן $\Pi(3) = 1$.

וכן הלאה.



תרגיל

חשבו את פונקציית הרישא לתבנית

aaabaaba

תרגיל

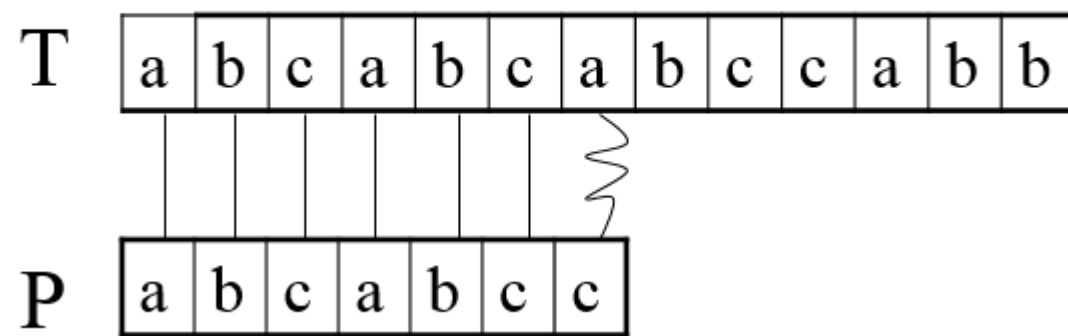
חשבו את פונקציית הרישא לתבנית

aabaaba

תשובה:

i	0	1	2	3	4	5	6	7
P	-1	0	1	0	1	2	3	4

דוגמה:



הרעיון הכללי של האלגוריתם: מחפשים את P ב-T. בכל פעם שיש אי-התאמה, נסיט את P ימינה בהיסט הקטן ביותר האפשרי שמקיים שעדיין יש התאמה של התווים הראשונים ב-P לבין התווים האחרונים ב-T. האלגוריתם:

דוגמת הרצה:

j :	0	1	2	3	4	5	6
$p[j]$:	a	b	a	b	a	a	
$b[j]$:	-1	0	0	1	2	3	1

0	1	2	3	4	5	6	7	8	9	...
a	b	a	b	b	a	b	a	a		
a	b	a	b	a	c					
	a	b	a	b	a	c				
		a	b	a	b	a	c			
			a	b	a	b	a	c		
				a	b	a	b	a	c	

הגדרה-פונקציית הסיפא

$\sigma(i) = j - \text{הוא אורך הרישא המקסימלית של התבנית } P_j \text{ שהיא סיפא של } T_i \text{ (לכל } 0 < i \leq n \text{)}$.

לדוגמא:

T = a a a b a a c a a

	0	1	2	3	4	5	6	7	8	9
σ	0	1	2	2	3	4	5	0	1	2

P = a a b a a

נחפש ב- $T_4 = a a a b$ את הסיפא המקסימלית שהיא רישא של $P_3 = a a b$.

נחפש ב- $T_6 = a a a b a a$ את הסיפא המקסימלית שהיא רישא של $P_5 = a a b a a$.

הערה: ניתן לחשב את פונקציית
הסיפא בעזרת KMP

פסאדו קוד

נחשב את $T(s)$ ונתאחל $i=j=0$

כל עוד $i \leq |T| - |S|$:

אם $T(i+j) = S(j)$

אם $j = |S| - 1$ נחזיר שמצאנו ונעדכן את החיפוש.

אחרת $j++$ ונעדכן את פונקציית הסיפא. $\sigma(i+j) = j$.

אחרת:

$i += j - T(j)$

$j = \max(0, T(j))$

תרגיל:

הגדרה: נסמן ב- T^R את המחרוזת ההופכית של T (באורך n), כלומר $T^R = T(n-1)T(n-2)\dots T(0)$.

T פלינדרום אם $T = T^R$.

בהינתן מחרוזת T באורך n , תארו אלגוריתם שמוצא את אורך הרישא המקסימלית של T שהינה פלינדרום.

$T = a b a a b a b a$

T_0 —
 T_2 ———
 T_6 —————

לדוגמא-

אז נחזיר $j=6$.

פיתרון:

אם x רישא של T , אז קיים y כך ש- $T = xy$. לכן מתקיים $T^R = y^R x^R$.

אם x פלינדרום אז $T^R = x^R$ (גם הכיוון השני נכון).

לכן מספיק למצוא את הרישא הארוכה ביותר של T שהיא סיפא של T^R .

← נריץ KMP כך הטקסט שלנו הוא T^R והתבנית היא T ונחזיר $\sigma(n) = j$ המקסימלית כך ש- T_j סיפא של T_n^R .

סיבוכיות: $O(n) = O(2n) = O(m+n)$. (נאיבי: $O(n^2)$).

תרגיל:

נתונה מחרוזת T שאורכה $n \geq 10$. תארו אלגוריתם המוצא חלוקה $T = xy$ כך ש- $|y| \geq 10$ והאורך של x מקסימלי.

*הערה – תמיד קיימת חלוקה כזו, כיוון ש- x יכולה להיות ε .

$$T = \underline{aaba} \overbrace{aaaaabbcbbaaaa}^{|y|=11} \underline{aaba}$$

דוגמא:

$$x = adUda$$

פיתרון:

מה יכול להיות האורך המקסימלי של x ? $|x| \leq \left\lfloor \frac{(n-10)}{2} \right\rfloor = k$

• בדוגמא : $k = \left\lfloor \frac{21-10}{2} \right\rfloor = 5$

רוצים למצוא את הרישא המקסימלית של T באורך לכל היותר k שהיא גם סיפא של T .

← נריץ KMP כאשר התבנית היא T_k והטקסט הוא k התווים האחרונים ב- T .

סיבוכיות: $O(n)$.
דוגמא –
 $T = \overset{P}{\underline{aabaada}} \overset{T}{\underline{abbcccbba}} \underline{caabaa}$

```
caabaa
| | | |
aabaada
```

$$|x| = 5$$

BM

האלגוריתם משווה את התבנית מול תחילת הטקסט אך מתחיל את ההשוואה של מסוף התבנית עד לתחילתה. האלגוריתם עושה preprocessing על התבנית ויוצר 2 פונקציות. מטרתן לקדם את התבנית תוך כדי "דילוג" על תווים שאנחנו יודעים בודאות שהתבנית לא תימצא בהם.

- BAD CHAR TABLE
- GOOD SUFFIX TABLE

BM-BST

הרעיון: במידה וטעינו בתו- כמה ניתן ללכת כדי לתקן.
במידה וטעינו בתו שקיים בתבנית- נרצה להתאים את התו המתאים בתבנית.
במידה וטעינו בתא שלא קיים בתבנית- נרצה לדלג לגמרי על התו.

המשך-BST

- טבלת BST - מקבלת את התו במחרוזת T בו הייתה אי התאמה. נסמן $m = \text{len}(P)$ ו $n = \text{len}(T)$.

$$BST[x] = \begin{cases} m & x \notin P \\ m - i - 1 & x = P_{m-1}[i] \end{cases}$$

לדוגמה - אם $P = abcba$
 $P_{m-1} = abcb$

$$BST[x] = \begin{cases} 4 & x = a \\ 1 & x = b \\ 2 & x = c \\ 6 & \text{else} \end{cases}$$

GST

נגיד וטעינו במקום ה-i-.

נרצה להגדיר את GST להיות:

1. המחרוזת הכי ימנית שחופפת לסיפא אחרי ; ושונה מ si.
2. הרישא הארוכה ביותר שחופפת לסיפא של הסיפא הנ"ל.

לדוגמה:

טעינו כאן BANACANABANA

המחרוזת CANA חופפת ל ANA ושונה מב !

עוד דוגמה:

BANACANABANA

BANA חופפת לסיפא של ABANA.

תרגיל

הגדרה: מחרוזת T' היא סיבוב מעגלי של מחרוזת $T = t_1 t_2 \dots t_n$ אם קיים $1 \leq i \leq n$ כך שמתקיים:

$$T' = t_i t_{i+1} \dots t_n t_1 t_2 \dots t_{i-1}$$

דוגמא: $car \leftrightarrow arc$

נתונות 2 מחרוזות T, T' באורך n . תארו אלגוריתם הבודק האם T' הינה סיבוב של T .

פתרון

פיתרון:

נשים לב ש-T' סיבוב מעגלי של T אם"ם T מופיע ב-TT.

$$TT = t_1 t_2 t_3 \dots t_{i-1} \underbrace{t_i t_{i+1} t_{i+1} \dots t_n t_1 t_2 \dots t_{i-1}}_{T'} t_i \dots t_n$$

← נריץ KMP עם טקסט TT ותבנית T'.

סיבוכיות: $O(2n+n) = O(n)$.