

## מבני נתונים ואלגוריתמים - הרצאה 19

10 בינואר 2012

### אלגוריתם KMP

דוגמה:

$j$	0	1	2	3	4	5	6	7	8	9	10
$W$	A	C	T	A	C	T	A	C	A	G	T
$C(j)$	-1	0	0	0	1	2	3	4	5	1	0

האלגוריתם למציאת  $C$  (קצת שונה מההרצאה הקודמת, הקודם היה קצת לא נכון):

#### אלגוריתם 1 אלגוריתם למציאת $C$ באלגוריתם KMP

```

C(0) = -1
C(1) = 0
i=1
j=0
while (i < N) // N is the length of W
{
  while (W(i) = W(j))
  {
    C(i+1)=C(j) + 1;
    i++;
    j++;
  }
  if (C(j) ≠ 0)
    j = C(j)-1;
  else
  {
    C(i+1) = 0;
    i++;
    j=0;
  }
}

```

### אלגוריתם (Boyer Moore) BM

האלגוריתם הזה, בניגוד לאלגוריתם הקודם, מסתכל לא רק איפה טעינו ב  $W$  אלא גם מה היה מולו ב  $T$ . הבדיקה באלגוריתם זה היא מסוף המילה לתחילתה. גם פה יש לנו טבלה, שבה מוחזק המקום המאוחר ביותר שמופיעה כל אות במילה - Bad Shift Table = BST. לדוגמה, במילה BANANA:

אות	כמה אפשר להסיט - BST
B	5
A	0
N	1
אחרים	6

נאתחל שני אינדקסים -  $i = k - 1$  שיסמן את המיקום בטקסט ו  $j = 0$  שיסמן את ההסטה אחורה. עושים השוואה בין  $T(i - j)$  לבין  $W(k - 1 - j)$ .

אם יש אי התאמה, נעשה:  
 $i = i + BST(T(i - j)) - j$   
 $j = 0$

אם יש התאמה:  
 $i = i$   
 $j++$

אך במקרה זה יכולה להיות בעיה, שבה  $j > BST(T(i - j))$  כשאין התאמה ואז  $i$  יורד, ואנו לא מתקד-מים בבדיקה.

לדוגמה, במילה BANANA, אם כשבדקנו את ה- $N$  המאוחר יותר מצאנו  $A$ , נתקדם ב-1.

לכן, יוצרים גם טבלה שנקראת Good Shift Table - GST. בטבלה הזו, לכל אות בודקים כמה אפשר להסיט קדימה אם טעינו באות זו, לפי כך שיש את האותיות שבדקנו כבר ונכונות אבל האות שבדקנו ולא נכונה היא אחרת מהאות שבדקנו.

W	B	O	N	A	N	A
GST	6	6	6	2	6	1
W	B	A	N	A	N	A
GST	6	6	2	6	4	1

נשים לב שאם יש לנו רישא של כל המילה שהיא גם סיפא של כל המילה, יש קצת בעיה עם איך שמצאנו עד עכשיו, כי הכי הרבה שנוכל לקדם את המילה זה שהרישא תשב מול הסיפא:

W	B	A	N	A	B	A
GST	4	4	4	4	2	1

לכן התיקון שנעשה לאלגוריתם הוא שכאשר אין התאמה בבדיקה בין  $T(i - j)$  לבין  $W(k - 1 - j)$ , נעשה:

$$i = \max(i + BST(T(i - j)) - j, i + GST(W(k - 1 - j)))$$

#### אלגוריתם 2 אלגוריתם BM

```

i=k-1
j=0
while (i < L)
{
  while (W(k - 1 - i) == T(i - j) && j < k)
    j++;
  if (j == k)
    cout "success"
  i = i + max(BST(T(i - j)) - j, GST(k - 1 - j))
  j=0
}

```