

# מבני נתונים

מבוסס על הרצאותיו של פרופ' שמואל קליין

## שיעור 1

## שיעור 2

### אלגוריתם Boyer & Moore

אלגוריתם לחיפוש תת מחרוזת שפותח ב-1977 ע"י Bob Boyer and J Strother Moore. החידוש בו הוא שבעזרת האלגוריתם שלהם,  $C = 1$ .

בהינתן 2 מערכים,  $T, P$ , כאשר אורכו של  $T$  הוא  $n$  ואורכו של  $P$  הוא  $m$  ( $n > m$ ), נרצה לחפש את  $P$  בתוך  $T$ . החידוש באלגוריתם הוא שהחיפוש נעשה מהסוף להתחלה.

$\Delta_1$  מייצג את המרחק המינימלי שבו ניתן "להזיז" את התאמת תת המחרוזת (למעשה, גודל תת המחרוזת), מבלי לפספס מופעים קודמים. איך נדע שלא קיים מופע נוסף של האותיות בטקסט? נשתמש בטבלת עזר: לכל תו בתת המחרוזת שנחפש, יותאם הערך המקסימלי שמייצג את מרחקה מהדלטה ע"פ הנוסחה (המספר מייצג נסיגה).  $i \leftarrow \Delta_1(T[i])$ . לכל שאר האותיות - נאתחל אותם להיות  $\Delta_1$ .  $T[i]$  הוא מיקום המופע האחרון של האות המבוקשת. אך לעיתים המספר המינימלי  $T[i]$  הוא קטן מדי במקרה של אי-התאמה וניתן לדלג  $k + 1$  צעדים (כאשר  $k$  הוא מס' הצעדים שנסוגנו בחיפוש תת המחרוזת). לכן ניתן לתקן את הנוסחה להיות

$$i \leftarrow \max(\Delta_1(T[i]), k + 1)$$

נגדיר  $\Delta_2$  להיות היוריסטיקה על המרחק המינימלי שבו נמצאת תת המחרוזת, (להשלים) ולכן נתקן את הנוסחה להיות  $i \leftarrow \max(\Delta_1(T[i]), k + 1, \Delta_2(T[j]))$

## שיעור 3

### הקצאה

sequential \ dynamic allocation  
postfix, prefix, infix

תור, מחסנית, דו-תור  
תיאור אלגוריתם postfix:

```
read(x);
if x notin {operators}
    Push(x);
else {
    y <- Pop();
    z <- Pop();
    w <- Compute(z, x, y);
    Push(w);
}
```