# מבני נתונים ואלגוריתמים - הרצאה 21

17 בינואר 2012

## בניית עץ סיפא

נחלק לכמה פעולות:

### הוספת אות עבורה אין מסלול

- מוסיפים ענף חדש מהשורש

### הוספת אות לאורך מסלול כשהיא כבר קיימת במסלול

- לא עושים כלום

### הוספת אות לאורך מסלול כשהיא לא קיימת במסלול

- תיקון העץ - בהמשך

אלגוריתם 1 אלגוריתם לבניית עץ סיפא - עם 2 הפעולות הראשונות (בלי השלישית)

יש לנו מילה באורך $m$, $ap$ שהוא מצביע למיקום נוכחי בעץ, $ed$ שהיא נק' העצירה, ו$j$ שהוא עומק בענף הנוכחי.
יש לנו עץ, בו כל קדקד יכול להכיל מס' ילדים כמס' האותיות, וכל קדקד מכיל זוג מספרים.
כל קדקד פנימי מכיל Suffix Link לרצף שמסתיים באותו מקום אבל מתחיל אות אחת יותר מאוחר (NANA מחזיק ANA).

```
ap = head
ep = head
j=0
for i=0 ... m-1 // m = length of word
{
  if ap doesn't have a child equal to W(i) and j=0:
  {
    add a new child to active point with value (i, ..)
  }
  k = ap[1] // ap's value is (k, ..)
  else if w(k + j)==w(i)
  {
    j++
  }
  else if w(k + j) ≠ w (i)
  {
    Repair_Tree
  }
}
```

כעת נכתוב את הפונקציה Repair_Tree:

```
while j>0
{
  ap.value = (ap.value[1], k+j-1)
  add a child to ap with value (i, ..)
  if ap has no child which fits to w (k + j), add a son with value (k+j, ..)
  if ap has no suffix link
  {
    find ap's suffix
add a link from ap to the suffix
    ap = suffix
    j–
  }
  if we arrived to a vertex which doesn't need repairing, make it the active point and break.
}
```