

```

Build-SuffixTree(S, n):
S[0,1,...,n-1] // מחרוזת
AP = EP = root
K=0
// לולאה שעוברת על כל התווים במחרוזת
for i=0 to n-1
    if AP == root and S[i] לא קיים
        צור ענף חדש עם S[i]
    else if AP == root and S[i] קיים
        כנס לענף המתאים
        K++
        AP מצביע למקום החדש
    else // AP is not root
        if S[AP+k] == S[i]
            נכנסים "פנימה" באותו ענף
            K++
        else // S[AP+k] is not S[i]
            AP = Repair-Tree(AP, k)
        end
    end
end
end

```

AP = Repair-Tree(AP, k):

הוספת הסיפא  $S[i-k:k]$ :

פצל את הקודקוד עליו מצביע AP ל:  
 אב:  $(AP, AP+k-1)$  בן ימני:  $(i, )$  בן שמאלי:  $(AP+k, )$   
 K--

```

while k > 0 // לולאה להוספת הסיפוא  $i-k:k$ 
    חיפוש מיקום של הסיפוא הבאה  $S[i-k:k]$  ו-EP יצביע למיקום המתאים
    EP = Search-Suffix(S[i-k:k], AP)
    if k==0
        ניצור Suffix-Link מ-AP לשורש (כי קודקוד האב של AP הוא בהכרח השורש
        ונשאר לנו רק תו יחיד להוסיף, לכן רישא של תו יחיד זה מחרוזת ריקה).
        AP → root
    else
        אם לא קיים עדיין, ניצור Suffix-Link מ-AP ל-EP
        AP → EP
    end
    אם  $S[i-k:i]$  קיים and  $K==0$ 
        במקרה זה, בגלל  $k=0$ , אז אנחנו בעצם רק רוצים להוסיף את התו  $S[i]$ , אז אם קיים
        ענף שמתחיל ב- $S[i]$ , פשוט נכנסים אליו, כמו בפונקציה הראשית, ומקדמים בחזרה
        את K ב-1, ויוצאים מהלולאה.
        כנס לענף המתאים
        K++
        AP מצביע למקום החדש
        Break
    else if
        אם  $S[i-k:i]$  אינו רישא של אחד הענפים שיוצאים מ-EP
        צור ענף חדש עם  $S[i-k:i]$ 
    else

```

עבור כל קודקוד (פרט לשורש), צריכים לשמור מי הקודקוד הראשון שיוצא מהשורש בענף של אותו הקודקוד. נקרא לו  $FN$  (First Node=).

```
פצל את הקודקוד עליו מצביע EP ל:
אב: (EP, FN+k-1) בן ימני: (i, ) בן שמאלי: (FN+k, )
end
AP = EP
end
```

$P = \text{Search-Suffix}(S, P)$ :

הרעיון הכללי: אם יש Suffix-Link, הרי שהוא מצביע לאותה הסיפא, פחות התו הראשון – וזו הסיפא שאנחנו מחפשים.

אם אין, עולים למעלה בעץ, עד שמוצאים Suffix-link, ואז נרד כמספר הצעדים שעלינו. אם הגענו לשורש (כלומר במסלול אחורה לא היה בכלל Suffix-link), יורדים בענף המתאים ומחפשים ת המיקום המתאים להכנסת הסיפא הנוכחית.

```
while flag == false
  if P == root
    P מתקדם לקודקוד הבא בענף המתאים
  if there exists a Suffix-Link from P
    P = Suffix-Link מגיע ה-Suffix-Link.
  else // not root and no suffix link - going up in tree
    P = parent(P)
  end
  if P מצביע על המיקום המתאים של S
    flag = true
  end
end
return P
```