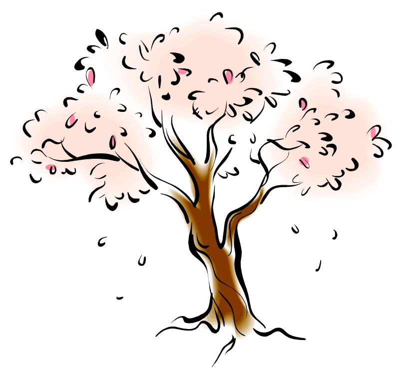
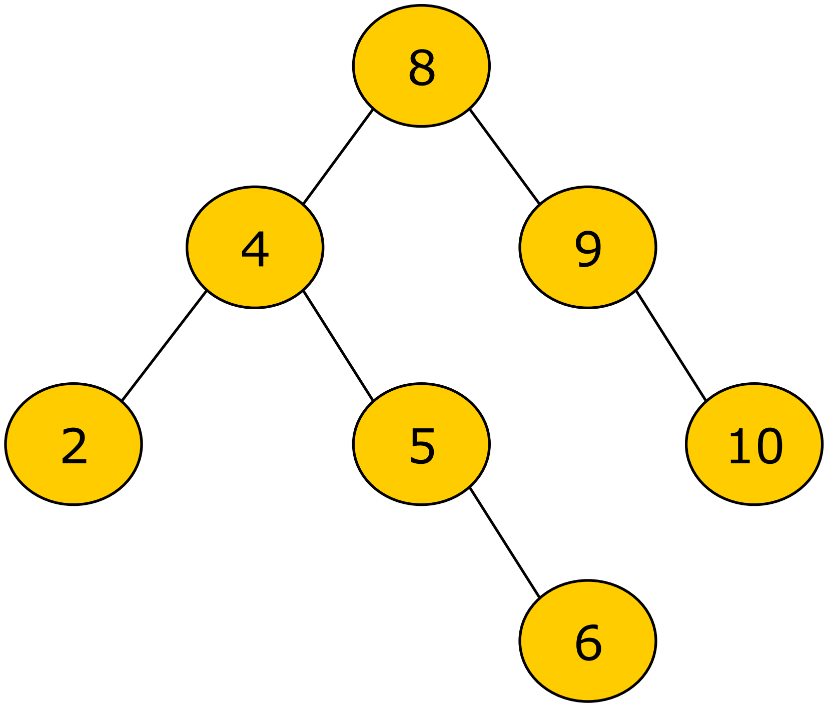
מבני נתונים

תרגול 2



ליאור שפירא

עצי חיפוש בינאריים



- לכל קודקוד 4 שדות
  - left – מצביע לבן שמאלי
  - right – מצביע לבן ימני
  - p – מצביע לאב
  - key - מפתח

כל המפתחות בתת עץ ימני > v.key > כל המפתחות בתת עץ שמאלי

תרגיל

- בהינתן עץ חיפוש בינארי (בעל n צמתים), הדפיסו את כל המפתחות בו, בסדר ממוין, בזמן O(n)

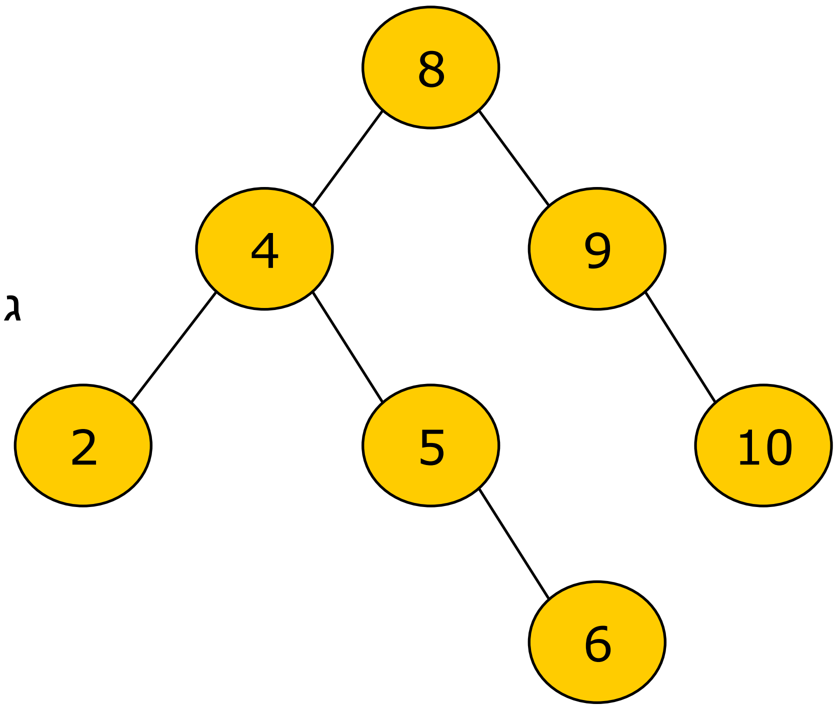
<u>In-order(v)</u>

```
If (v==null) return
Else
    In-order(v.left)     ?  גודל תת עץ שמאלי
    Print v.key          ?  זמן קבוע (1)
    In-order(v.right)    ?  גודל תת עץ ימני
    Return
End if
```

תרגיל

- בהינתן עץ חיפוש בינארי (בעל n צמתים), הדפיסו את כל המפתחות בו, בסדר ממוין, בזמן O(n)

```
In-order(v)
If (v==null) return
Else
    In-order(v.left)     ?  גודל תת עץ שמאלי
    Print v.key          ?  זמן קבוע (1)
    In-order(v.right)    ?  גודל תת עץ ימני
    Return
End if
```



ההוכחה: באינדוקציה, עוברים על כל צומת פעם אחת ולכן O(n)

תרגיל Amortized

- הראו כיצד לממש ADT של מחסנית (לא מוגבלת)בעזרת מערך, כך שכל פעולה תיקח Amortized O(1) וגודל המערך תמיד יהיה O(n) כאשר n מס' האברים במחסנית
  - אנו נראה פתרון בו גודל המערך קטן שווה ל-4 פעמים מס' אברי המחסנית
- הפתרון
  - נגמר מקום במערך ← מגדיל פי 2
  - נשארו ½ מהאברים ← נקטין פי 2

לא טוב, ייתכן מצב בו נגדיל נקטין כל הזמן, יקר!

- פתרון אחר
  - כשנגמר המקום נגדיל פי 2
  - כשהמערך מלא רק עד כדי ¼ נקטין פי 2

מימוש הפתרון

- Size – מס' האברים במחסנית
- Max-size – גודל המערך
- A – מערך בגודל Max-size

Initialization

```
Size ← 0
Max-size ← 2
A ← array of size 2
```

Push(S,x)

```
If (size<max-size)
    A[size+1] ← x
    Size ← size + 1
Else
    A' ← new array of size 2*max-size
    A ← A' (+copy)
    max-size ← max-size*2
    A[size+1] ← x
    Size ← size +1
End if
```

Pop(S)

```
Return-value = A[size]
Size ← size-1
If (size < max-size/4)
    A' ← array of size max-size/2
    A ← A' (+copy)
    Max-size ← max-size/2
End if
```

המשך התרגיל...

- משפט: עלות ה-Amortized היא O(1)
  - פעולת הכנסה יקרה ← המערך ½ מלא
  - פעולת הוצאה יקרה ← המערך ½ מלא
- למה 1 – תמיד אחרי פעולה "יקרה" המערך ½ מלא
- למה 2 – נניח op1 פעולה יקרה, ו-op2 הפעולה היקרה אחריה. נניח שאחרי op1 גודל המערך k אזי בין op1 ל-op2 יש לפחות k/4 פעולות
- הוכחה
  - אחרי Op1 גודל המחסנית הוא m/2
  - אזי, אם op2 פעולת הגדלה, אז עד אליה יהיו ≥ m/2 פעולות
  - אם op2 פעולת צמצום, אז עד אליה יהיו ≥ m/4 פעולות. מ.ש.ל

הוכחת המשפט (שיטת הבנק)

- תזכורת:
  - amortized(op) = cost(op) – withdrawal + deposit
- נרצה להוכיח שכל פעולה amortized(op) = O(1)
- במקרה שלנו:
  - כל פעולה תפקיד 4 מטבעות
  - פעולה יקרה תמשוך מטבעות בגודל המערך שהיא מקצה
  - נרצה להראות שהפעולות הזולות משלמות על הפעולה היקרה שבאה אחריהן
- פעולת הגדלת מערך
  - עולה m (העתקה של m)
  - לפניה היו לפחות m/4 פעולות שהפקידו 4, כלומר $4 \cdot {}^{m}\!/\!{}_{4} = m$
  - לכן, הבנק תמיד בפלוס, ועלות כל פעולה מכוסה

נדלג על שאר ההוכחה (הקטנת המערך)

הוכחת המשפט (שיטת הפוטנציאל)

- נרצה להגדיר פונק' פוטנציאל Φ כך ש:
  - Φ(start)=0
  - Φ(end)≥0
- תזכורת: $amortized(op) = \cos t(op) + \Delta\Phi$
- נרצה להוכיח שלכל פעולה $amortized(op) = O(1)$
- נגדיר את Φ:

$$\Phi = (\begin{cases} \frac{\max size}{2} - size & \text{if } size < \frac{\max size}{2} \\ 2(size - \frac{\max size}{2}) & \text{if } size > \frac{\max size}{2} \end{cases}) - 1$$

בהורדה

בהוספה

בעזרת Φ ניתן לעשות דברים "עדינים" יותר משיטת הבנק