

(List) רשימה

דף 21 (1) list-anchor - הקשר בין הפרטים הפרטניים.

(ח) הנכם נשאלים - מנחים את האיברי שכתב לעמקין ו אגרי כרמל, שידו חזק  
 להנחים בנשימה 3, 5, 7 את הסדה 4 בין 5/5, 5/5, 5/5, 5/5  
 את 4 לעמקין שף 3 והוא יכנס אתו.

(ח) הוצא מרשימה - מקומה מקור של איבר במשימה (מקום) אותו.

(א) עיקר הדרישה - מנהל המבחן מקיים הדרישה (מנהל המבחן) את האחריות

(1) קופס בבטיחה.  $\phi$  הבט / הקופס. גרמניה.

0(1) זמן - list-end - במקום גיטו אתרי פאטין.

(10) אמר מרשמה - מנצח את האמר פטרקוס הרבן פרייטער.

(ד) לכבן רשעים. מקובל אומר ומקור ומעשרה את האביר במקום זה האביר הנק.

(?) הסיבה חקירה? ~~מחשבה~~ אם מחשבה צמח אס פרשימה חקירה מאוהרים.

0(9) אחרי השנה - מחזור השני חיקה.

முடிவுரை

רשימה סה פאיר הארין נצח'ס זאגן

(Stack) نچو

לומר היא משהי מחנך, מחנך, אלה חנכו

last in First out, ~~הראשון~~ שבראשון הוא הראשון שצאת.

~~החלטות~~ (Top) - האישיות החזקה

○  $\text{is\_empty}()$  -  $(\text{Create Empty Stack})$  -  $\text{is\_empty}()$  -  $\text{is\_empty}()$

১৭) পুষ্টি - শক্তি

פרק 10 - (Pop) - גודל אוכלוסייה  $O(1)$

4) מעריך חזרה (IsEmpty) - מודד אם קיים איבר בריבוע.

①) הוא ממונה (stackTop) - ממונה את הזרק של כל המסמכים (שם) (שם)

## תור (Queue)

ראש התור (Front) - הנגזר מהתור

סוף התור - סוף התור

סוגי תורים: FIFO, First in First out, ה- ראשון שנסקף הראשון יצא.

תור (Init) - יצירת תור

הכנס לתור (Enque) - הכנסת איבר לסוף

הוצא מתור (Deque) - הוצאת איבר מהראש

הראש התור (Head) - איבר ראשון בתור

האם תור? (Is-Empty) - האם תור?

## תור כפול (Double Ended Queue - DEQ)

הכנסת איבר לתור (Enque)

ניתן להכנס לתור משני הצדדים ולקבל את התוצאה בשני הצדדים.

תור (Create) - יצירת תור

הכנס לתור (Insert-Head) - הכנסת איבר לראש התור

הכנס לתור (Insert-Tail) - הכנסת איבר לסוף התור

הוצא מתור (Delete-Head) - הוצאת איבר מהראש

הוצא מתור (Delete-Tail) - הוצאת איבר מהסוף

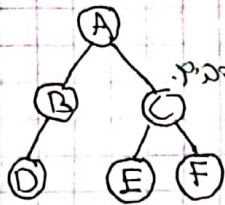
קבלת איבר (Read-Head) - קבלת איבר מהראש

קבלת איבר (Read-Tail) - קבלת איבר מהסוף

האם תור? (Is-Empty) - האם תור?



# עץ (Tree)



עץ אינו כולל ציורים שונים של אותו קבוצת קשרים.

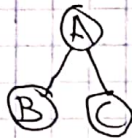
כמו כן אינו נקרא ציור, לאיבר הקבוצה קוויים אלא

ואפואותם הנ"ל ציורים שהם מאתו אלא נקראים אגפים.

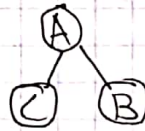
האיבר בתוספת העץ (עץ אחד) נקרא גזע ואילו הצמתים נקראים ענפים.

צורה של צומת היא מספר הנ"ל שיש לצומת.

עץ מוצק - הוא עץ בעל גישות לצומת:



$\neq$



עץ בינארי - עץ צומת יש עץ ביותר 2 הנ"ל

עץ בינארי מלא הוא עץ בעל צומת צורה 2 בפיו.

עץ בינארי מלא - הוא עץ בינארי מלא בו כל הענפים באותה רמה.

בנות של עץ בינארי

מה עץ  $(L, R, X)$  - מאתם עץ בינארי ומעבר כאשר הסדר  $X$  יהיה ימני.

תת-עץ של  $(T)$  - בעזרת המעבר את התת-עץ השמאלי של  $T$ .

תת-עץ ימני  $(T)$  - בעזרת המעבר את התת-עץ הימני של  $T$ .

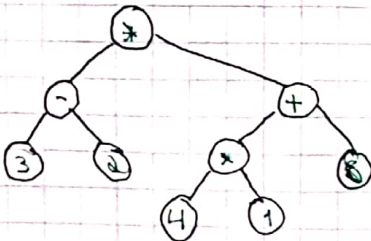
הגוף תת-עץ ימני/שמאלי  $(T)$  - בעזרת המעבר את התת-עץ הימני/שמאלי של  $T$ .

אזור שבה  $(T)$  - מעבר את השורש של  $T$ .

ענף שבה  $(T, X)$  - בעזרת המעבר את התכן של השורש  $T$  -  $X$ .

עץ חיק  $(T)$  - מעבר את  $T$  עץ-חיק או שרר אגרת.

מאפיין גזעני באמצעות עץ-בינארי



$$((3-2) * ((4+1) + 8))$$

מילון ממשק אנושי של השמות המכונים (מפתח, אינדקס, וכו').  
באופן כללי כל המידע שנרשם במילון הוא מידע אודות המפתח האנציקלופדי הממוצע  $U$ .

פעולות

- איתור -  $Find(D, x)$  - הגדר מובנית למילון  $D$  שבה  $x$  או לא.
- הוספה -  $Insert(D, x, w)$  - הוסף  $x$  למילון  $D$  עם משקל  $w$ .  
*המשקל*
- הוצאה -  $Delete(D, x)$  - הוצא  $x$  מהמילון  $D$  שבו הוא נמצא.

כדורים

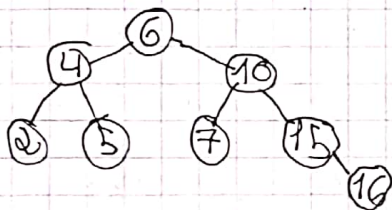
$x \in U \leftarrow x$  גודל (כדור)

פעולות (מבנה) סופי יש סדר  $U$  -  $U$   
(מבנה סופי)

- מינימום -  $min(D)$  - המינימום במילון  $D$ .
- מקסימום -  $max(D, x)$  - המקסימום במילון  $D$  שבו  $x$  הוא המינימום.  
*מקסימום*
- המילון הוא מבנה של מילון עם מפתח  $x$  וצורה  $U$ .

המילון  
מבנה

- כל המבנה הוא מבנה של מילון עם מפתח  $x$  וצורה  $U$ .
- כל המבנה הוא מבנה של מילון עם מפתח  $x$  וצורה  $U$ .



Preorder - 6, 4, 2, 5, 10, 7, 15, 16

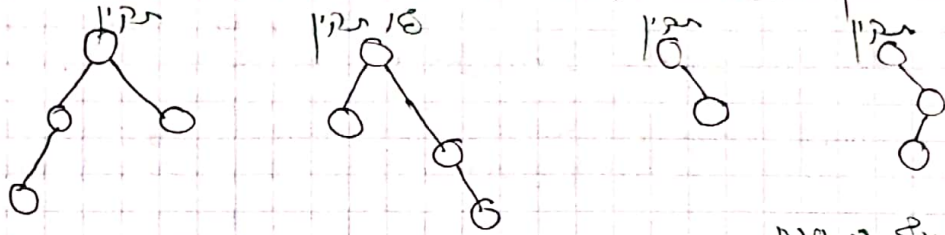
InOrder - 2, 4, 5, 6, 7, 10, 15, 16

PostOrder - 2, 5, 4, 7, 16, 15, 10, 6



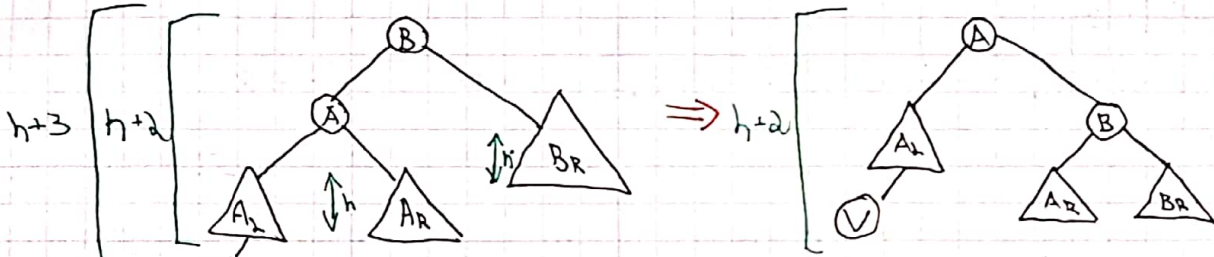
# AVL - מאזן-פ-ז

AVL מאזן-פ-ז הוא סוג של מאזן-פ-ז המבטיח כי גובהו של העץ הוא  $O(\log n)$ , כלומר, העץ הוא מאוזן. היתרון של AVL הוא שהוא מאזן את העץ באופן אוטומטי, כלומר, אין צורך במאזן-פ-ז ידני.



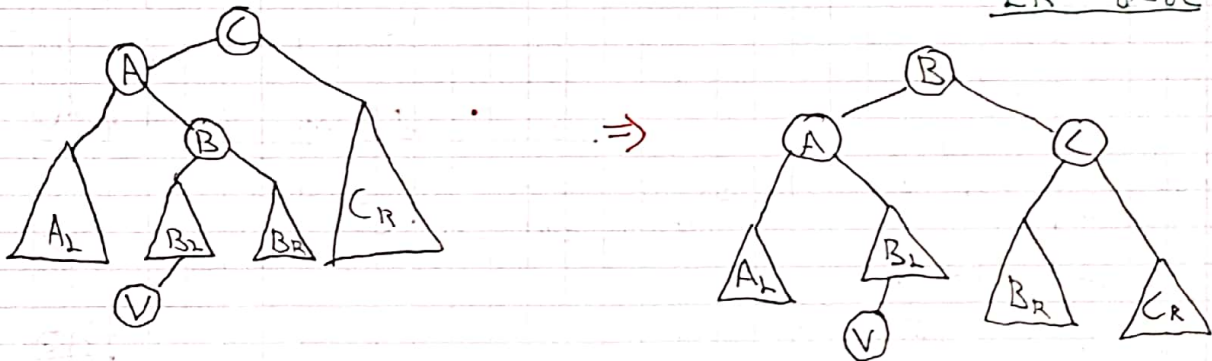
אילונו של AVL

אם יש הפרש בגובה בין שני צדדים, נבצע איזון. ישנן שתי דרכים לבצע איזון:  $LR$  (שמאל-ימין) ו- $RL$  (ימין-שמאל). היתרון של  $LR$  הוא שהוא פשוט יותר מבין השניים.



\* האיזון  $AR > A$  יעקור ע"י  $LR$ . איזה שיטה של  $LR$  או  $RL$  תבצע? תלוי במבנה העץ.

LR



# תור עדיפות - Priority Queue

תור עדיפות הוא תור עם מנגנון העדפת הנתונים שבהם עדיפותם שונה.

→ כל איבר מתווכת עדיפות נאומה.

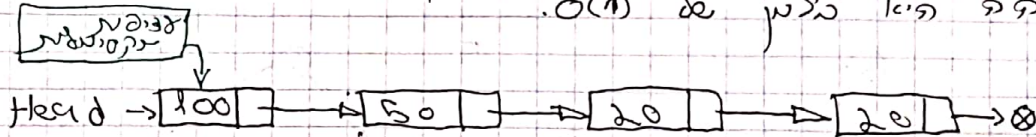
→ סדר התקנת איברים הוא שרירותי.

→ האישים מתווכת עדיפות הם סדר עדיפות - האיבר הראשון שאיפה הוא האחרון שהתווכת בתור.

\* אנו השימוש העיקרי ביותר עובר עדיפות הוא ניהול משימה של CPU, עדיפות משימה של עדיפות (עדיפות) הן נתונים עדיפות.

## ממשל תור עדיפות על פני רשימה

נתון רשימה מקושרת ממנו"ת בסדר יורד של עדיפות, כך שהעדיפות הגבוהה ביותר תהיה הראשונה והעדיפות הנמוכה היא בתחתית של  $O(1)$ .



## פעולות

$O(1)$  הכנסה - רצונך למצוא מקום עם סדר עדיפות

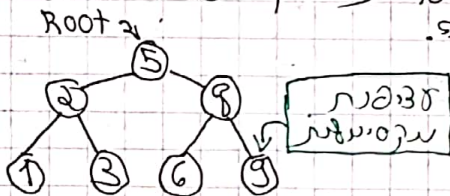
$O(1)$  העברה - של האיבר העדיף

$O(1)$  העברה - של האיבר העדיף

$O(1)$  מציאה - של האיבר העדיף

## ממשל של תור עדיפות על פני עץ

נתון עץ גבוה בינארי המיון עדי עדיפות כך שהעדיפות הגבוהה תהיה בתחתית העץ.



## פעולות

$O(\log n)$  הכנסה - של נתון גבוה

$O(\log n)$  העברה - של האיבר העדיף

$O(1)$  מציאה - של האיבר העדיף (עץ מבוזבז שמתווכת יציביות אולי)

## ממשל של תור עדיפות על פני עדיפות

עדיפות (heap) הוא מבנה נתונים המאפשר מציאת איבר מקסימלי במהירות.

## פעולות

יציאת עדיפות -  $O(1)$

הכנסה -  $O(\log n)$

מציאת מקסימום -  $O(\log n)$

מציאת מינימום -  $O(1)$

יציאת עדיפות על פני  $Heap(H, value)$  -  $O(n)$



8 במה (המשק)

סגולת דמיון (המשקל)



# רשימה קופצית (Skip-List)

```
typedef struct node {
```

```
    key value;  
    struct node *next, *down;
```

```
} NODE;
```

```
NODE* new-node (key k, NODE* n, NODE* d)
```

```
{
```

```
    NODE* p;  
    p = (NODE*) malloc (sizeof(NODE));  
    p->value = k;  
    p->next = n;  
    p->down = d;  
    return p;  
}
```

רשימת קופצית היא מבנה נתונים של רשימה מקושרת אשר כל שם מספר נתון, כל מספר אחרים נוספים קיימים. כל שם נתון איבר ברשימה של וכל המבנה הקרוי רשימה קופצית (בדרך כלל קוראים לה כך כי אינה שלישית).

פונקציה

הוספה  $O(\log n)$  - הוספה של איבר חדש לרשימה. הוספה של איבר חדש לרשימה, גודל רשימה אחת או יותר, וכל המבנה של המבנה החדש, גודל המבנה החדש.

החלפה  $O(\log n)$  - החלפה של איבר אחד באתר אחר. החלפה של איבר אחד באתר אחר, החלפה של איבר אחד באתר אחר.

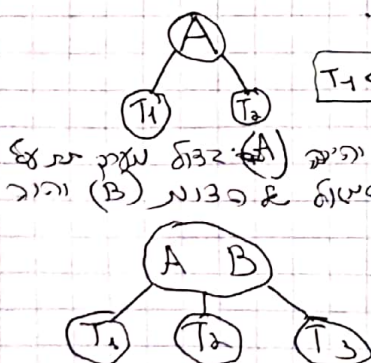
החלפה של איבר אחד  $O(1)$  - החלפה של איבר אחד באתר אחר, החלפה של איבר אחד באתר אחר.

החלפה של איבר אחד  $O(\log n)$  - החלפה של איבר אחד באתר אחר, החלפה של איבר אחד באתר אחר.

## רשימת קופצית - דוגמה

כל המבנה הוא רשימה קופצית. כל המבנה הוא רשימה קופצית.

המבנה של המבנה הוא רשימה קופצית. המבנה של המבנה הוא רשימה קופצית.



$$T_1 < A \leq T_2$$

$$T_1 < A \leq T_2 < B \leq T_3$$

פונקציה

Find(p, x)  $O(\log n)$

Delete(n, x)  $O(\log n)$

min(D)  $O(1)$   
max(D, x)  $O(1)$

