רשימות דילוגים
וסיבוכיות משוערכת

Skip Lists רשימות דילוגים

חומר קריאה לשיעור זה

"Skip lists: A probabilistic Alternative to Balanced Trees", William Pugh, Communications of the ACM, 33(6): 668-676, 1990.

המאמר נמצא גם באתר הקורס תחת skip list animation וכן באתר:
www.cs.umd.edu/users/pugh (Research/Papers)

Skip Lists רשימות דילוגים

עצים מאוזנים, שנלמדו בשני השעורים הקודמים, משמשים למימוש פעולות המילון (חיפוש, הכנסה, הוצאה) בזמן $O(\log n)$ במקרה הגרוע ביותר.

מימוש הפעולות אינו טריוויאלי, במיוחד כאשר יש צורך לתמוך גם בפעולות הדורשות לשמור בכל צומת אינפורמציה ייחודית (כמו rank) ולעדכן אותה.

נלמד כעת מבנה נתונים בעל מימוש פשוט מאוד, שלא דורש איזונים של מבנה הנתונים לאחר הכנסה\הוצאה. בתמורה: בצוע פעולות המילון יעשה בזמן $O(\log n)$ בממוצע.

כמו כן ההסתברות שזמן בצוע פעולה יחרוג בצורה משמעותית (נאמר פי 3 מהזמן הממוצע היא זניחה (פחות מ- 1 ל 100,000,000).

מהו "ממוצע" ?

ראינו כבר (בשיעור 3) שעצי חיפוש ללא פעולות איזון יכולים לשמש למימוש פעולות המילון בזמן $O(\log n)$ ב"ממוצע".

מהו ממוצע? בניתוח שעשינו הנחנו שהוכנסו n איברים בסדר אקראי. קיימות $n!$ פרמוטציות, ומכאן סדרי הכנסה. הממוצע של גובה $n!$ העצים שנוצרו הוא $O(\log n)$.

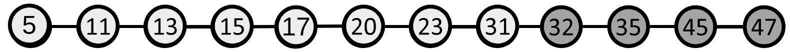
הבעייתיות בניתוח זה נובעת ממספר סיבות:

- ראשית, בד"כ ההסתברות להופעת פרמוטציה בקלט אינה אחידה.
- שנית, מבנה הנתונים שנוצר תלוי בסדר הכנסת הנתונים: "יריב" (adversary) - אדם\תהליך המחבל במערכת - יכול להכניס סדרת נתונים כך שהפעולות יבוצעו בזמן האיטי ביותר.

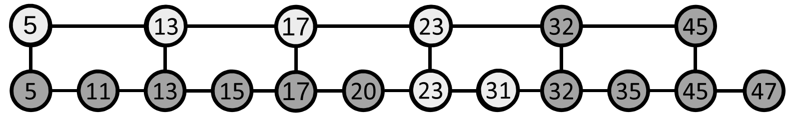
ברשימות דילוגים בעיות אלה נפתרות. הפעולות וזמן ביצוען תלוי בהגרלות שעושה המחשב ולא בסדר הכנסת הנתונים. הממוצע מחושב לפי ההגרלות ולא לפי הנחות על ההסתברות של הקלטים. אלגוריתם כזה נקרא **אלגוריתם רנדומלי**.

הרעיון המרכזי

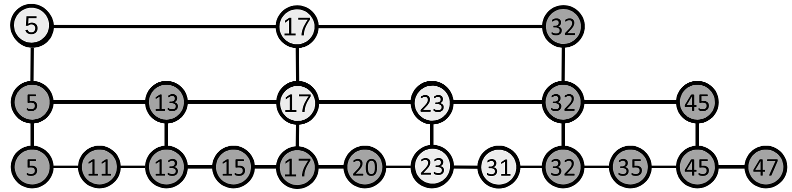
חיפוש איבר בסוף רשימה הוא יקר.



כדי לזרז את החיפוש (פי שניים) נוסיף מדריך-תת רשימה של כל איבר שני (הצמתים הבהירים הם הצמתים שמשתתפים בחיפוש אחר 31):



כדי לזרז את החיפוש במדריך נוסיף רמה נוספת:

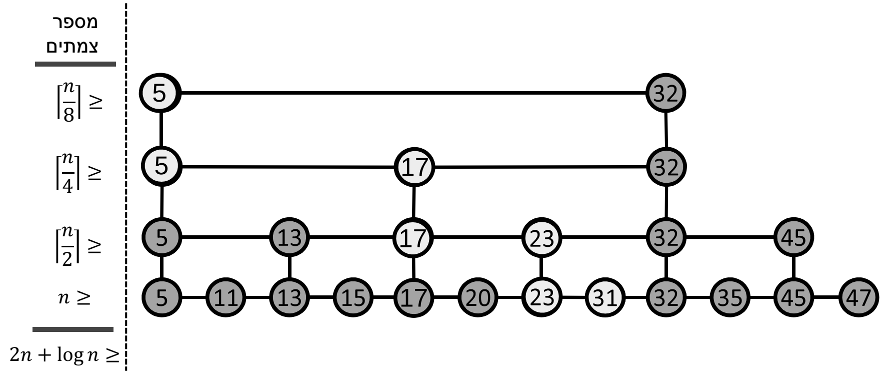


וכך הלאה עד לרמה $\lceil \log n \rceil$ ובה איבר בודד (ניתן לוותר על רמה זו).

סיבוכיות המקום

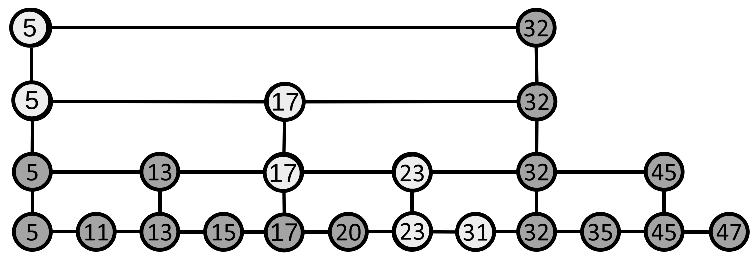
סיבוכיות מקום: אם מספר האיברים הוא n אז מספר האיברים ברמה השניה הוא $\lceil \frac{n}{2} \rceil$, ברמה הבאה $\lceil \frac{n}{4} \rceil$ וכך הלאה. לפיכך סה"כ מספר הצמתים הוא לכל היותר

$$n + \left\lceil \frac{n}{2} \right\rceil + \left\lceil \frac{n}{4} \right\rceil + \cdots + 1 \le n + \left(\frac{n}{2} + 1\right) + \left(\frac{n}{4} + 1\right) + \cdots \le 2n + \log n = O(n)$$



סיבוכיות זמן חיפוש

סיבוכיות זמן החיפוש: נסתכל על מסלול החיפוש מצומת הסיום ועד צומת ההתחלה. במעבר מרמה לרמה אנו עוברים על פני 2 מצביעים לכל היותר. לפיכך אורך המסלול הוא לכל היותר $2\lceil \log n \rceil = O(\log n)$.



הערה: נשים לב שאלגוריתם החיפוש נכון גם אם מספר הצמתים של רמה i בין כל שני צמתים של רמה $i-1$ אינו קבוע.

בעיה: כיצד ניתן לשמור על המבנה שיצרנו בעת הכנסה והוצאה מבלי שיהיה צורך לארגן את כל הרמות מחדש?

תשובה: נטיל מטבעות!

הכנסה באמצעות הטלת מטבע

צעדים להכנסת מפתח k.

1. חפש את k. אם k נמצא, סיים.
2. שמור מצביע לצומת הימני ביותר בכל רמה במסלול החיפוש.
3. הוסף צומת חדש ברמה התחתונה ביותר וקבע את המפתח להיות k.
4. לפי סדר הרמות מלמטה למעלה:

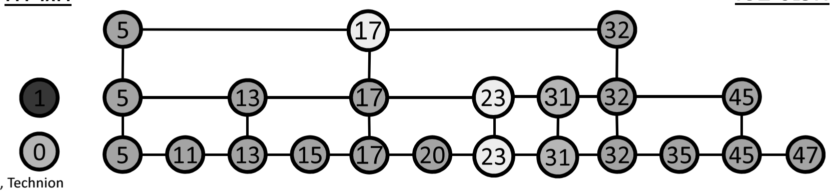
toss( ) - הגרל מטבע.

אם יוצא 0: הוסף צומת חדש מעל הרמה הנוכחית וקבע את המפתח בו להיות k. אם ברמה העליונה הוגרל 0, הוסף רמה חדשה.

אם יוצא 1: עצור.

הכנס 31:

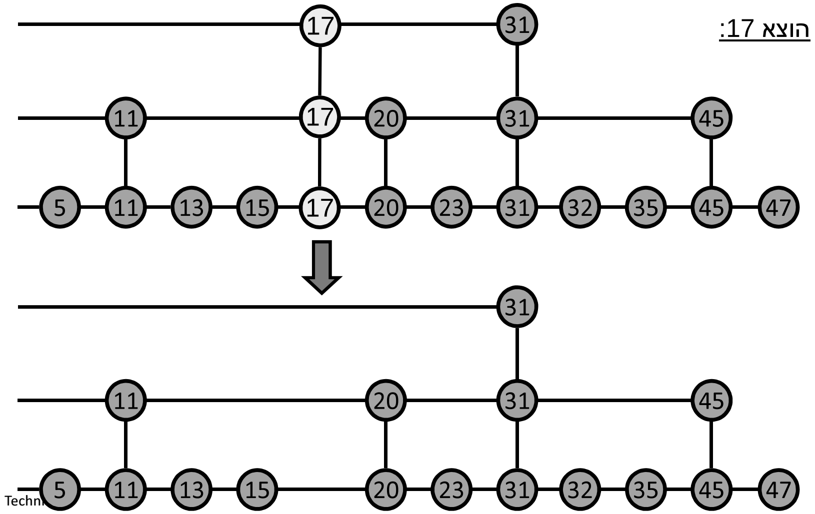
הגרלה



הוצאה מרשימת דילוגים

1. מצא את האיבר בעל המפתח k.
2. הוצא איבר זה מכל הרמות בהם הוא מופיע.

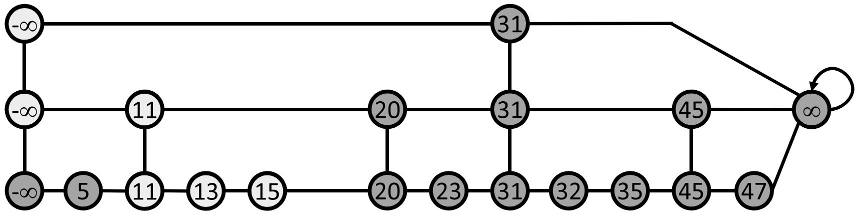
<u>הוצא 17:</u>



הערות לגבי מימוש רשימת דילוגים

נוח להוסיף בכל רמה צומת אחד בהתחלת הרשימה שמפתחו $-\infty$ וצומת אחד בסוף שמפתחו $+\infty$.

<u>מצא 17:</u> לא נמצא.



תוכניות החיפוש, ההכנסה וההוצאה ניתנות לכתיבה רקורסיבית וגם לא רקורסיבית בקלות רבה. דוגמאות למימוש <u>בחוברת השקפים</u>.

תוכניות לרשימת דילוגים

<u>הגדרת צומת ופרוצדורת עזר ליצירתו:</u>

```
typedef  struct  node {
            Key  value ;
            struct node  *next,  *down ;
            } NODE ;
NODE  *top ;
```

value	next
	down

```
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 ;
}
```


הערות לגבי מימוש

תזכורת: נוח להוסיף בכל רמה צומת אחד בהתחלת הרשימה שמפתחו $-\infty$ וצומת אחד בסוף שמפתחו $+\infty$.

תוכנית ההכנסה רקורסיבית אך ניתן לכתוב אותה גם ללא רקורסיה בקלות רבה. באתר הקורס תחת skip list animation קיים קוד לא רקורסיבי בשפת C++ ובשפת Java. יש הבדלים מסוימים בין התוכניות השונות.

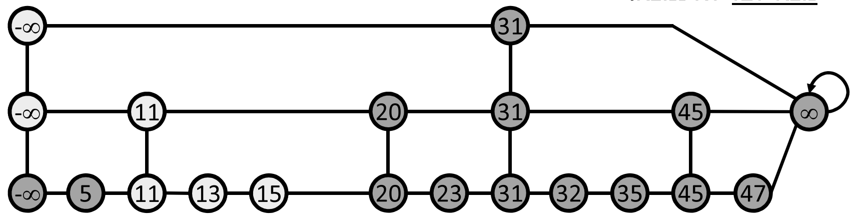
<u>הנחות והסברים קצרים של פרוצדורות במימוש בעמודים הבאים:</u>

- הפרוצדורה add_after(k,p) מוסיפה צומת חדש בעל מפתח k אחרי p, ומחזירה מצביע אליו.
- הפרוצדורה toss( ) מחזירה 0 או 1 בהסתברות שווה.

תוכנית חיפוש ברשימת דילוגים

```
NODE   *search (KEY k, NODE  *p) {
        /* returns a pointer p to last node ≤ k  */
        while(1) {
                for ( ; p → next →  value  <= k ; p =  p → next) ;
                if ( p → down = = NULL)  break;   /* final level */
                p = p → down ;
        }
        return p ;
}
```

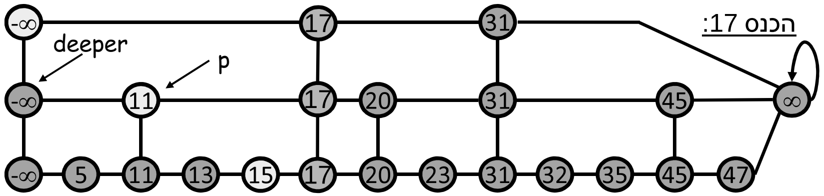
מצא 17: לא נמצא.



תוכנית להכנסה ברשימת דילוגים

```
NODE   *insert1 (KEY k, NODE  *p) {
   /* returns a pointer to the added node */
   NODE * deeper, *new_node;
   for ( ; p → next!=NULL && p → next →  value  <= k ; p =  p → next) ;
   if ( p → down = = NULL)  return add_after(k,p) ;          /* final level */
   deeper = insert1(k, p → down);
   if (( deeper == NULL || toss( ) ) then { return  NULL};
   else {  new_node = add_after(k,p) ;
           new_node → down = deeper ;}
  return new_node;
}
```

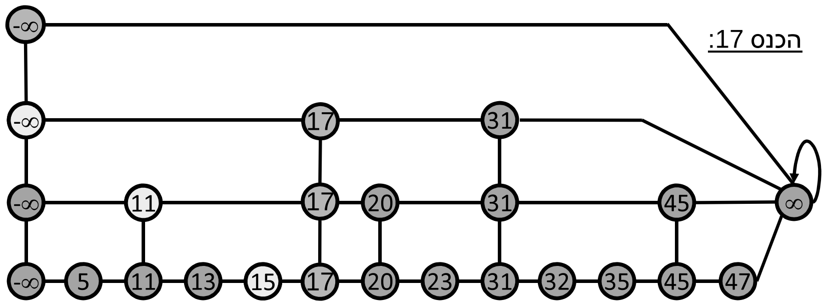
הכנס 17:



הכנסה ברמה העליונה

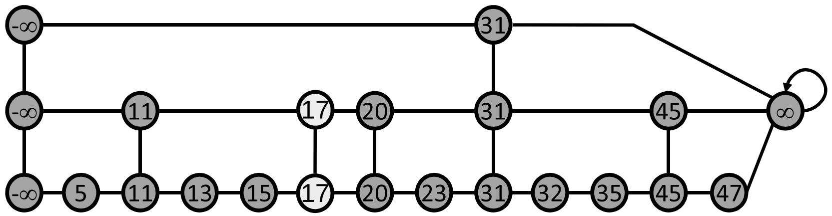
```
NODE   *insert (KEY k, NODE  *old_list){
     /* returns a pointer to the added node */
     NODE * deeper ;
     deeper = insert1(k, old_list);
     if (( deeper == NULL || toss( ) ) return  old_list ;
     return   new_node(-INFINITY, inf-addrs, old_list);
}
```

הכנס 17:



ניתוח מקום

- בכל רמה האלגוריתם מטיל מטבע כדי לקבוע אם יוצר העתק נוסף מעליה.
- מספר הרמות הממוצע למפתח שווה למספר הפעמים הממוצע שיש להטיל מטבע הוגן עד שיצא 1.
- מספר הפעמים הממוצע שיש להטיל מטבע הוגן עד שיצא 1 הוא 2. (הוכחה בשקף הבא. אינטואיציה: בממוצע מטבע נופל פעם על צד 0 ופעם על צד 1).



מסקנה: מספר הצמתים הממוצע הוא $2n$, כאשר n הוא מספר המפתחות במבנה (כל מפתח מופיע פעמיים בממוצע). מכאן, המקום הנדרש הוא $O(n)$ בממוצע.

הטלת מטבע

נניח שלמטבע הסתברות p לצאת "זנב" (נסמן צד זה באפס) והסתברות $1-p$ לצאת "ראש" (נסמן צד זה באחד).
מהו T, מספר הפעמים שיש להטיל את המטבע, בממוצע, עד שנקבל "ראש" ?

נסמן ב- Q_i את ההסתברות שנקבל "ראש" לראשונה בהטלה ה- i (נקרא התפלגות גאומטרית).
<u>מתקיים:</u> $Q_i = p^{i-1}(1-p)$.

$$T = \sum_{i=1}^{\infty} i \cdot Q_i = \sum_{i=1}^{\infty} i \cdot p^{i-1}(1-p) = (1-p)\sum_{i=1}^{\infty} i \cdot p^{i-1} = \frac{1-p}{(1-p)^2} \quad (|p|<1)$$

כלומר בממוצע נדרשות $\boxed{T = \frac{1}{1-p}}$ הטלות עד שמקבלים "ראש".

ועבור מטבע הוגן, כאשר $p = \frac{1}{2}$, נדרשות $T = 2$ הטלות בממוצע.

ניתוח זמן

<u>משפט:</u> L, האורך הממוצע של מסלול חיפוש ברשימת דילוגים עם n מפתחות, מקיים: $L \leq 2\log_2(n) + 2$.

<u>הוכחה:</u> נסתכל על מסלול חיפוש מנקודת הסיום אחורנית לנקודת ההתחלה. המסלול מורכב מתנועה מעלה (כשאפשר) ותנועה שמאלה (כשחייבים). כאשר מגיעים לקצה השמאלי ביותר, התנועה ממשיכה כלפי מעלה בלבד.

נסמן ב- $c(k)$ את אורך המסלול הממוצע המטפס k רמות.
מתקיים: $c(0) = 0$
$c(k) \leq ½\,(1 + c(k)) + ½\,(1 + c(k-1))$
<u>הסבר:</u> צעד שמאלה בהסתברות ½ (אם אפשר) וצעד מעלה בהסתברות ½.

לפיכך: $c(k) \leq 2 + c(k-1)$

ולכן: $c(k) \leq 4 + c(k-2) \leq \cdots \leq 2k$

כלומר אורך מסלול ממוצע מרמה 1 לרמה $\log_2(n)$ הוא לכל היותר $2(\log_2(n) - 1)$.

אורך מסלול חיפוש (המשך)

<u>המשך ההוכחה:</u> נותר לקחת בחשבון את המקרים שבהם מספר הרמות גדול מ- $\log_2(n)$.
מעל הרמה ה- $\log_2(n)$ מספר הצעדים הממוצע שמאלה חסום ע"י מספר האיברים הממוצע ברשימת הדילוגים שהוכנסו מעל רמה זו. מספרם הממוצע חסום ע"י 2 שכן בכל רמה מספר האיברים הממוצע קטן פי 2.
מהו מספר הצעדים כלפי מעלה?

ניתן להראות שמספר הרמות המקסימלי הממוצע עבור רשימת דילוגים עם n איברים קטן מ- $\log_2(n) + 2$.
כלומר מספר הצעדים הממוצע כלפי מעלה מעל הרמה $\log_2(n)$ הוא לכל היותר 2.

לפיכך סך אורכו הממוצע של מסלול חיפוש, L, מקיים: $L \leq 2(\log_2(n) - 1) + 4$ כמצוין במשפט.

זמן בצוע הפעולות

<u>מסקנה מהמשפט:</u> חיפוש, הכנסה, והוצאה מרשימת דילוגים נעשים בזמן ממוצע $O(\log n)$, כיון שבכל צעד על מסלול החיפוש\הכנסה\הסרה מתבצעים $O(1)$ צעדים.

זמן בצוע הפעולות

<ins>מסקנה מהמשפט</ins>: חיפוש, הכנסה, והוצאה מרשימת דילוגים נעשים בזמן ממוצע $O(\log n)$, כיון שבכל צעד על מסלול החיפוש\הכנסה\הסרה מתבצעים $O(1)$ צעדים.

<ins>הערה</ins>: כאשר משתמשים במטבע עם הסתברות לקבלת "זנב" p, אזי מספר ההעתקים הממוצע לכל מפתח הוא $\frac{1}{1-p}$ ומספר הרמות הממוצע הוא $\log_{1/p}(n)$.

<ins>הכללת המשפט</ins>: האורך הממוצע של מסלול חיפוש, L, ברשימת דילוגים עם n מפתחות כאשר למטבע הסתברות p לצאת "זנב" מקיים:
$L \le 1/p \cdot \log_{1/p}(n) + 1/p$. (כלומר 2 מוחלף ב- $1/p$).

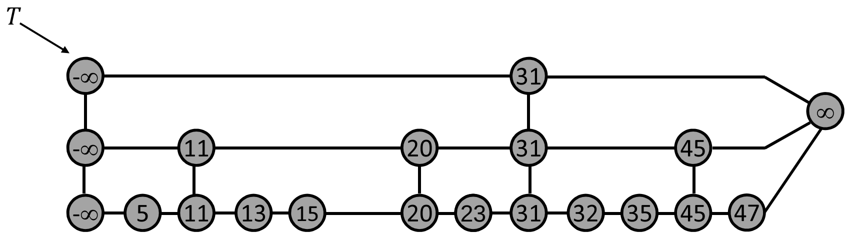
לפרמטר p אין השפעה גדולה על האורך הממוצע של מסלול חיפוש.
פרקטית מומלץ לבחור $p=\frac{1}{4}$, <ins>אם</ins> להסתברות לחריגה משמעותית מהממוצע אין חשיבות רבה.
<ins>התוצאה</ins>: מרווח ממוצע של 4 בכל רמה. אחרת, רצוי לבחור מטבע הוגן: $p=\frac{1}{2}$.

רשימת דילוגים דטרמיניסטית
Deterministic Skip List

מבנה אחר הקשור לרשימת דילוגים אקראית היא רשימת דילוגים <ins>דטרמיניסטית</ins>.

מבנה נתונים זה דומה לרשימת דילוגים (אקראית) אותה ראינו בהרצאה זו, אך מאפשר פעולות חיפוש\הכנסה\הסרה בזמן זהה לעצים מאוזנים. דהיינו, בזמן $O(\log n)$ <ins>במקרה הגרוע</ins>.

עוד על מבנה זה בתרגולים.



שיטות אנליזה לניתוח זמנים משוערך
Amortized Time Analysis

חומר קריאה לשיעור זה

Chapter 17– Amortized Analysis (405 – 429)

חלק מהשקפים משתמשים בתרגומיו הנאים של המתרגל גיל כהן בקורס שיטות באנליזה של אלגוריתמים (236715) משנת 2008.

זמן ומחיר משוערך

<u>הגדרה</u>: אם m פעולות מתבצעות בזמן כולל $T(m)$, אזי <u>הזמן המשוערך</u> לכל פעולה מוגדר להיות $T(m)/m$.

<u>עניין בהגדרה</u>: כאשר ישנה סדרה ארוכה של פעולות המדד המעניין הוא סך זמן הריצה, ולא עלות פעולה בודדת. אם רוב הפעולות קלות לביצוע וחלקן קשה לביצוע, אז ההשפעה של הפעולות הקשות מתחלקת על סדרת הפעולות כולה.

בהרצאה זו נמיר את מושג הזמן במושג מחיר. נדבר על <u>מחיר משוערך ומחיר בפועל</u>. המרה זו רלוונטית כל עוד המחיר הכולל פרופורציונאלי לזמן הכולל. המרה פשוטה זו מאפשרת חשיבה אנליטית בזווית ראיה חדשה.

<u>פורמלית:</u> אם נבצע פעולות m שעולות $c_1, c_2, \ldots, c_m$ בהתאמה, נרצה לחסום מלמעלה את $T(m) = \sum_{i=1}^{m} c_i$.

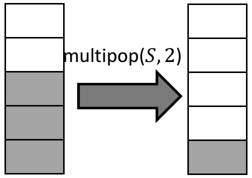
(זאת בניגוד למטרתנו עד כה, לחסום את מחיר הפעולה היקרה, כלומר $\max_{1\le i\le m} c_i$).

<u>נראה שלוש שיטות ניתוח:</u> צבירה, חיובים, פוטנציאל.

מחסנית עם פקודת הוצאה מורחבת

נזכר בפעולות הרגילות של המחסנית push(S,x), pop(S) הלוקחות זמן $O(1)$.
נגדיר את מחיר כל פעולה להיות יחידה אחת.

<u>נגדיר פעולה נוספת:</u>
multipop(S,k) הוצא מהמחסנית את k האיברים העליונים. אם ישנם פחות מ-k איברים, רוקן את המחסנית.



<u>המימוש</u>: ע"י בצוע פעולות pop(S) בלולאה עד להתרוקנות המחסנית או הוצאת k איברים. זמן $O(\min(k,s))$ כאשר s הוא מספר האיברים במחסנית בעת ביצוע הפעולה. מחיר הפעולה מוגדר כמספר ההפעלות של פעולת ה pop(S).

ניתוח בשיטת הצבירה
Aggregate Analysis

<u>בשיטת הצבירה</u> לעריכת אנליזה משוערכת מראים ישירות שכל סדרה של m פעולות עולה במקרה הגרוע מחיר כולל $T(m)$, ומכאן שהמחיר המשוערך של פעולה במקרה הגרוע הוא $T(m)/m$.

<u>בניתוח גס:</u> פעולת multipop בודדת יכולה לעלות מחיר m ולכן במקרה הגרוע עבור סדרה של m פעולות יידרש מחיר של $O(m^2)$.

<u>בניתוח עדין</u>: לכל סדרה של m פעולות יידרש במקרה הגרוע מחיר $O(m)$, שכן כל איבר שמוצא היה חייב להיות מוכנס. מכאן שכאשר מתחילים ממחסנית ריקה המחיר הכולל של m פעולות הוא $O(m)$. כלומר, המחיר המשוערך של פעולה הוא $O(1)$.
תוצאה זו היא למקרה הגרוע ביותר. היא נכונה לכל סדרת פעולות באורך m.

ניתוח בשיטת החיובים
Accounting Method

<u>בשיטת החיובים</u> זוקפים לחובתה של כל פעולה עלות הנקראת <u>המחיר המשוערך</u> a_i (amortized cost). במקרים בהם המחיר המשוערך של פעולה גבוה מהמחיר בפועל, c_i, ההפרש משמש כיתרת זכות (credit). ביתרה זו ניתן להשתמש בעתיד לסיוע בתשלום פעולות שהמחיר המשוערך שלהם קטן ממחירים בפועל.

אם לא ניכנס לחוב, הרי שסך התשלומים הוא חסם עליון על סך המחיר בפועל. כלומר,

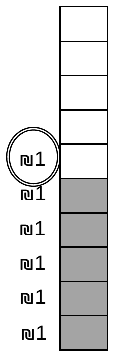
$$\sum_{i=1}^{m} a_i \ge \sum_{i=1}^{m} c_i$$

ניתוח המחסנית בשיטת החיובים

המחיר בפועל לפעולות המחסנית הוא:
- 1 שקל לפעולת pop( ).
- 1 שקל לפעולת push(x).
- min(k, s) שקלים לפעולת ה-multipop(k).

נגדיר מחיר משוערך של 2 שקלים לכל פעולת push(x)
ומחיר 0 שקלים עבור שאר הפעולות.

<u>כיצד נדאג לא להיכנס לחוב?</u>
לכל push(x) אנו משלמים שקל אחד ומפקידים שקל אחד עבור הפעולה העתידית, שתוציא את האיבר הנ"ל מהמחסנית.
בעת הוצאה אנו פודים את אותו שקל שכבר צברנו, בעלות נוספת 0.



כיוון שלעולם לא ניכנס לחוב נקבל כי:

$$2m \ge \sum_{i=1}^{m} a_i \ge \sum_{i=1}^{m} c_i$$

והמחיר המשוערך לפעולה הוא אכן $O(1)$.

ניתוח בשיטת הפוטנציאל
The Potential Method

<u>בשיטת הפוטנציאל</u> נתייחס לתשלום מראש עבור הפעולות היקרות כאנרגיה פוטנציאלית של מבנה הנתונים כגוף אחד. פוטנציאל אשר ניתן לשחרר כדי לשלם עבור פעולות בעתיד.
נסמן את מבנה הנתונים לאחר פעולה i ע"י D_i וב- Φ את פונקציית הפוטנציאל.
בשיטה זו לא נקבע מראש את המחיר המשוערך. במקום זאת, המחיר המשוערך a_i נקבע ע"י המחיר בפועל, c_i, בתוספת הפרש הפוטנציאלים:

$$a_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

<u>בסיכום על כל הפעולות מקבלים:</u>

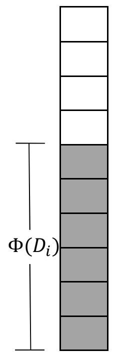
$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} c_i + \Phi(D_m) - \Phi(D_0)$$

במידה ולכל סדרת פעולות הפוטנציאל הסופי אינו קטן מהפוטנציאל ההתחלתי, כלומר $\Phi(D_m) \ge \Phi(D_0)$, נקבל כי המחיר המשוערך הכולל לסדרת הפעולות יהיה חסם עליון על המחיר הכולל בפועל של הסדרה.

נוח וגם ניתן לקבוע $\Phi(D_0) = 0$ ואז לוודא שלכל סדרה באורך m מתקיים $\Phi(D_m) \ge 0$.

ניתוח המחסנית בשיטת הפוטנציאל

הפוטנציאל Φ יוגדר להיות פשוט מספר האיברים במחסנית.
הפוטנציאל הוא 0 בתחילת הפעולות ולכל i הוא חיובי: $\Phi(D_i) \ge 0 = \Phi(D_0)$.



<u>בהגדרות אלה המחיר המשוערך של פעולה הוא:</u>
- עבור Push(x):
$$a_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = 1 + 1 = 2$$
- עבור multipop(S, k):
$$a_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = \min(s,k) - \min(s,k) = 0$$
- המחיר המשוערך ל-pop( ) הוא גם 0, מאותה סיבה.

<u>לכן בסיכום על כל הפעולות מקבלים:</u>

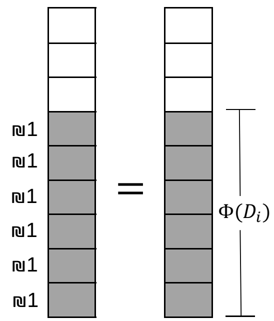
$$2m \ge \sum_{i=1}^{m} a_i = \sum_{i=1}^{m} c_i + \Phi(D_m) - \Phi(D_0) \ge \sum_{i=1}^{m} c_i$$

המחיר המשוערך ל-m פעולות הוא $O(m)$ וזהו חסם למחיר בפועל.

כסף כפוטנציאל
פוטנציאל ככסף

המחירים המשוערכים לפעולה שהתקבלו בניתוחים בעזרת שיטות החיובים והתשלומים עבור בעיית ה-multipop היו זהים: 2 ל- push(x), 0 לשאר הפעולות.
האם המצב מקרי?
<u>תשובה:</u> לא. ניתן לחשוב על הפוטנציאל כיתרת כסף ועל יתרת הכסף כפוטנציאל, כאשר למעשה ההבדל המרכזי בין השיטות הוא בבחירת הפוטנציאל\יתרה מראש או המחיר המשוערך מראש.

עבור הדוגמה הקודמת, קל להמחיש את שקילות השיטות באמצעות איור של יתרת הכסף והפוטנציאל:



שלוש השיטות ממעוף הציפור

1. <u>שיטת הצבירה:</u> מציאת חסם לכל סדרת פעולות באופן ישיר.
2. <u>שיטת החיובים:</u> צבירת קרדיט בגין פעולות מסוימות.
3. <u>שיטת הפוטנציאל:</u> צבירת פוטנציאל במבנה הנתונים לשימוש פעולות עתידיות.

$$\sum_{i=1}^{m} a_i = \sum_{i=1}^{m} c_i + \Phi(D_m) - \Phi(D_0)$$

כפי שציינו, שיטת הפוטנציאל והחיובים שקולות, אך לעתים יותר נוח להשתמש בשיטה אחת מאשר באחרת. לרוב שיטות אלו יותר גמישות משיטת הצבירה.

בעיית המונה הבינארי

נבחן את שלוש השיטות באמצעות פעולה על מונה בינארי בן k סיביות. ייצוגו של המונה הוא $A = a_{k-1}a_{k-2} \dots a_0$ וערכו הוא המספר המיוצג בו בבסיס 2, כאשר a_0 היא הספרה הפחות משמעותית. המונה מאותחל לאפס. הפעולה שנרצה לתמוך בה היא פעולת ה- increment המגדילה את ערך המונה באחד (מודולו 2^k).

$A = 00000000$
$A = 00000001$
$A = 00000010$
$A = 00000011$
$A = 00000100$
$A = 00000101$
...
$A = 11111111$
$A = 00000000$
$A = 00000001$
...

בעיית המונה הבינארי

הפעולה ממומשת באופן הבא:

```
increment(A) {
        i ← 0
        while (i < length(A) and a_i == 1) {
                a_i = 0;
                i = i + 1;
        }
        if (i < length(A))
                a_i = 1;
}
```

<u>לדוגמא:</u>
$A = 00100111$
$A = 00101000$

המחיר של פעולת increment תלוי ליניארית במספר הסיביות שהחליפו את ערכן. כיוון שיש רק k סיביות, בכל פעולה לכל היותר k סיביות יחליפו את ערכן, ולכן מחיר סדרה של m פעולות חסום ע"י $O(mk)$. לפיכך חסם פשוט למחיר (והזמן) המשוערך לפעולה הוא $O(k)$. אנו נראה כי ניתוח זה אינו הדוק.

ניתוח בשיטת הצבירה

<u>נתבונן בתחילת סדרת פעולות increment:</u>

$A = 00000000$
$A = 00000001$
$A = 00000010$
$A = 00000011$
$A = 00000100$
...

<u>נבחין כי לא כל הסיביות מחליפות את ערכן בכל קריאה ל-increment. יותר מכך:</u>

- סיבית a_0 מחליפה את ערכה בכל קריאה ל-increment.
- סיבית a_1 מחליפה את ערכה בכל קריאה שניה ל-increment.
- סיבית a_2 מחליפה את ערכה בכל קריאה רביעית ל-increment.
- ...
- סיבית a_i מחליפה את ערכה בכל 2^i קריאות ל-increment.

לפיכך המחיר הכולל חסום ע"י

$$T(m) \le \sum_{i=0}^{k} \frac{m}{2^i} < \sum_{i=0}^{\infty} \frac{m}{2^i} \le 2m = O(m)$$

במלים אחרות, המחיר המשוערך לפעולה הוא $O(m)/m = O(1)$.

ניתוח בשיטת החיובים

נגדיר מחיר משוערך של 2 שקלים לקביעת ערך של סיבית ל-1 ומחיר של 0 שקלים לקביעת ערך של סיבית ל-0.

<u>הרעיון:</u> כדי לא להיכנס לחוב, כאשר ערכה של סיבית נקבע ל-1 אנו משלמים שקל אחד מתוך השניים כדי להפוך את הסיבית. השקל השני נשמר כיתרת זכות עבור פעולת איפוס עתידית של סיבית זו.

	₪1		₪1		₪1	₪1			₪1
A	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**

במהלך ביצוע פעולת increment, סיבית אחת לכל היותר מחליפה את ערכה מ-0 ל-1, ולפיכך ניתן לשלם על הפעולה בעזרת לכל היותר 2 שקלים נוספים, כאשר על סיביות שמתאפסות נשלם בעזרת יתרת הזכות.

מכיוון שיתרת הזכות לעולם אינה שלילית (כל סיבית שמכבים הודלקה קודם) נובע שהמחיר המשוערך הכולל של m פעולות increment הוא $2m = O(m)$, וזהו חסם על המחיר הכולל בפועל.

כלומר, המחיר המשוערך לפעולה הוא $O(1)$.

ניתוח בשיטת הפוטנציאל

יהא b_i מספר ה-1-ים במונה. נגדיר את הפוטנציאל אחרי i פעולות increment, להיות מספר ה-1-ים במונה. כלומר, $\Phi(D_i) = b_i$.

אינטואיטיבית, הירידה במחיר ה-1-ים בעת פעולה בה מתהפכות הרבה סיביות אמורה לשלם את מירב מחיר הפעולה. נראה כי אינטואיציה זו אכן מוצלחת.

<u>נחשב את המחיר המשוערך של פעולת increment ה-i:</u>

- נסמן ב-z_i את מספר הסיביות שמתאפסות בפעולה ה-i.
- מספר הסיביות שמתחלפות הוא לכל היותר $z_i + 1$. לכן המחיר בפועל של הפעולה ה-i, c_i, מקיים: $c_i \le z_i + 1$.
- מספר ה-1-ים במונה לאחר הפעולה ה-i, b_i, מקיים: $b_i \le b_{i-1} - z_i + 1$.

$$A = \cdots 1010110\,\underbrace{111111111}_{z_i}$$

<u>לאחר הפעולה ה-i:</u>

$$A = \cdots 1010111\,\overbrace{000000000}$$

ניתוח בשיטת הפוטנציאל

<u>סיכום השקף הקודם:</u>

אם z_i מספר הסיביות שמתאפסות בפעולה ה-i אזי מתקיים:

- המחיר בפועל מקיים $c_i \le z_i + 1$.
- הפרש הפוטנציאלים מקיים

$$\Phi(D_i) - \Phi(D_{i-1}) = b_i - b_{i-1} \le 1 - z_i.$$

<u>מסקנה:</u> המחיר המשוערך קבוע:

$$a_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) \le (z_i + 1) + (1 - z_i) = 2$$

לסיום, נבחין כי הפוטנציאל ההתחלתי הוא 0 והפוטנציאל לאחר i פעולות לכל i תמיד גדול מאפס. לפיכך המחיר המשוערך הכולל של m פעולות increment מהווה חסם עליון על המחיר הכולל בפועל. חסם זה הוא $2m$. ומכאן העלות המשוערכת לפעולה היא אכן $O(1)$.