מבני נתונים ־ תרגולים

מבוסס על תרגולים של גלעד אשרוב, אונ' בר־אילן, 2012

שיעור 1

סדר גודל / אסימפטוטיקה / קצב גידול של פונקציות

נרצה לחשב סיבוכיות של בעיה, ניקח לדוגמה את מספרי פיבונצ'י המוגדרים ע"י נוסחת הרקורסיה: $\begin{cases} fib(1) = fib(2) = 1 \\ fib(n) = fib(n-1) + fib(n-2) \end{cases}$

```
int fib1(int n) {
        if (n == 1 || n == 2) {
                return 1;
        }
        return fib(n-1)+fib1(n-2);
}
```

```
int fib2(int n) {
        int preResult=1;
        result=1;
        temp=0;
        for(int i=2;i<n;i++) {
        temp=result;
        result+=preResult;
        preResult=temp;
        }
        return result;
}
```

חסם אסימפטוטי עליון ־ O (Big O notation)

הגדרה:

נאמר ש־ $f(n) \in O(g(n))$ אם קיימים שני קבועים $c > 0, n_0 > 0$ כך שלכל $n_0 < n \in \mathbb{N}$ מתקיים $f(n) \le c \cdot g(n)$

$$O(g(n)) = \{f(n) \mid \exists c, n_0 > 0, \forall n > n_0 \quad s.t. \quad f(n) \le c \cdot g(n)\}$$

הגדרה שקולה: נאמר ש־$f(n) \in O(g(n))$ אם קיים קבוע $c > 0$ כך ש: $\lim\limits_{n \longrightarrow \infty} \frac{f(n)}{g(n)} < c$

דוגמאות:

נניח שזמן ריצה של אלגוריתם A הוא $f(n) = 10n^2 + 5n$. נראה כי $f(n) \in O(n^2)$

לפי הגדרה 1:

$f\left(n\right)\leq c\cdot g\left(n\right)$**:**

$$10n^2+5n\underbrace{\leq}_{n_0=0}10n^2+5n^2=\underbrace{15}_{c}n^2$$

לפי הגדרה 2:

$$\lim_{n\longrightarrow\infty}\frac{f\left(n\right)}{g\left(n\right)}=\lim_{n\longrightarrow\infty}\frac{10n^2+5n}{n^2}=10$$

טענה: $n^2\notin O\left(n\right)$. מיידית לפי הגדרה 2: $\lim\limits_{n\longrightarrow\infty}\frac{n^2}{n}\longrightarrow\infty>c$ לכל קבוע c.

חסם אסימפטוטי תחתון - Ω (אומגה)

הגדרה: נאמר ש-$f\left(n\right)\in\Omega\left(g\left(n\right)\right)$ אם קיימים קבועים $c,n_0>0$ כך שלכל $n_0\leq n\in\mathbb{N}$ מתקיים $c\cdot g\left(n\right)\leq f\left(n\right)$

הגדרה שקולה: נאמר ש-$f\left(n\right)\in\Omega\left(g\left(n\right)\right)$ אם קיים קבוע $c>0$ כך שמתקיים $\lim\limits_{n\longrightarrow\infty}\frac{g(n)}{f(n)}\leq c$

דוגמאות: טענה: $3n^2+5\in\Omega\left(n^2\right)$

לפי הגדרה 1: $\begin{array}{c}c\cdot n^2\leq3n^2+5\\c\cdot g\left(n\right)\leq f\left(n\right)\end{array}$ והטענה נכונה עבור $c=1$, $n_0=1$.

לפי הגדרה 2: $\lim\limits_{n\longrightarrow\infty}\frac{n^2}{3n^2+5}=\frac{1}{3}$

טענה: $3n^2+5\notin\Omega\left(n^3\right)$, לפי הגדרה 2 אנו רואים מיידית כי $\lim\limits_{n\longrightarrow\infty}\frac{n^3}{3n^2+5}\longrightarrow\infty>c$ לכל קבוע c.

חסם הדוק אסימפטוטי - Θ

הגדרה: נאמר ש-$f\left(n\right)\in\Theta\left(g\left(n\right)\right)$ אם קיימים שני קבועים $c_1,c_2>0$ וקיים $n_0>0$ כך שלכל $n_0<n\in\mathbb{N}$ מתקיים: $c_1\cdot g\left(n\right)\leq f\left(n\right)\leq c_2\cdot g\left(n\right)$

הגדרה שקולה: נאמר ש-$f\left(n\right)\in\Theta\left(g\left(n\right)\right)$ אם קיים קבוע $c>0$ כך ש: $\lim\limits_{n\longrightarrow\infty}\frac{f(n)}{g(n)}=c$

דוגמה:

נראה כי $3n^2+5\in\Theta\left(n^2\right)$

לפי הגדרה 1:

$$\begin{array}{c}3n^2\leq3n^2+5\leq8n^2\\c_1\cdot g\left(n\right)\leq f\left(n\right)\leq c_2\cdot g\left(n\right)\end{array}$$

טענה: לכל שתי פונקציות חיוביות $f\left(n\right),g\left(n\right)$ מתקיים $f\left(n\right)\in O\left(g\left(n\right)\right)\wedge f\left(n\right)\in\Omega\left(g\left(n\right)\right)$ אםם $f\left(n\right)\in\Theta\left(g\left(n\right)\right)$

הסימון o ("o" קטן)

הגדרה: נאמר ש-$f\left(n\right)\in o\left(g\left(n\right)\right)$ אם לכל $c>0$ קיים $n_0\in\mathbb{N}$ כך שלכל $n_0\leq n\in\mathbb{N}$ מתקיים $f\left(n\right)<c\cdot g\left(n\right)$

הגדרה שקולה: נאמר ש- $f\left(n\right)\in o\left(g\left(n\right)\right)$ אם $\lim\limits_{n\longrightarrow\infty}\frac{f(n)}{g(n)}=0$

הסימון ω (אומגה קטן)

הגדרה: נאמר ש-$f\left(n\right)\in\omega\left(g\left(n\right)\right)$ אם לכל $c>0$ קיים n_0 כך שלכל $n_0\leq n\in\mathbb{N}$ מתקיים $c\cdot g\left(n\right)<f\left(n\right)$

הגדרה שקולה: נאמר ש- $f\left(n\right)\in\omega\left(g\left(n\right)\right)$ אם $\lim\limits_{n\longrightarrow\infty}\frac{f(n)}{g(n)}\longrightarrow\infty$

למעשה, O,Ω,Θ,o,ω הן מעין יחס סדר (גדול / קטן) :

$$\begin{array}{ccc}f\left(n\right)\in O\left(g\left(n\right)\right)&\approx&f\left(n\right)"\leq"g\left(n\right)\\f\left(n\right)\in\Omega\left(g\left(n\right)\right)&\approx&f\left(n\right)"\geq"g\left(n\right)\\f\left(n\right)\in\Theta\left(g\left(n\right)\right)&\approx&f\left(n\right)"="g\left(n\right)\\f\left(n\right)\in o\left(g\left(n\right)\right)&\approx&f\left(n\right)"<"g\left(n\right)\\f\left(n\right)\in\omega\left(g\left(n\right)\right)&\approx&f\left(n\right)">"g\left(n\right)\end{array}$$

מחלקות של פונקציות:

$$\begin{array}{rl} \text{Constants :} & \frac{1}{n^c} < \frac{1}{\log n} < 1 \ldots \\ \text{Logarithmic :} & \ldots < \log n < (\log n)^k < \ldots \\ \text{Polynomial :} & < \sqrt{n} < n < n \log n < n^k < \ldots \\ \text{Exponential :} & < 2^n < n! < n^n \end{array}$$

שאלה לדוגמה: בהינתן $f(n), g(n)$ האם תמיד מתקיים $f(n) \in O(g(n))$ או $g(n) \in O(f(n))$?
תשובה: לא. עבור $f(n) = n^{1+\sin n}$, $g(n) = n$ הטענה לא נכונה.

שאלה לדוגמה: האם לכל פונקציה $f(n)$ קיימת פונקציה סטנדרטית כך ש־$f(n) \in \Theta(g(n))$?
תשובה: לא. כמו בדוגמה הקודמת, עבור $f(n) = n^{1+\sin n}$, $g(n) = n$ הטענה לא נכונה.

תרגיל: מצא חסם אסימפטוטי הדוק לפונקציה $f(n) = \log(n!)$
ידוע כי $n! \le n^n$ ולכן $\log(n!) \le \log(n^n) = n \log n \in O(n \log n)$

$$\log(n!) = \log\left(\prod_{i=1}^{n} i\right) = \sum_{i=1}^{n} \log i \ge \sum_{i=\frac{n}{2}}^{n} \log i \ge \sum_{i=\frac{n}{2}}^{n} \log \frac{n}{2} \ge \frac{n}{2} \cdot \log \frac{n}{2} \in \Omega(n \log n)$$

מצאנו כי $f(n) = \log(n!) \in \Theta(n \log n)$

שיעור 2

נוסחאות נסיגה (רקורסיה)

מגדלי האנוי:

נגדיר בצורה רקורסיבית:
$$\begin{cases} T(n) = T(n-1) + 1 + T(n-1) \\ \qquad T(n) = 2T(n-1) + 1 \\ T(1) = 1 \end{cases}$$

פתרון נוסחאות נסיגה: שיטת האיטרציה, שיטת ההצבה (אינדוקציה), שיטת עצי רקורסיה

חיפוש בינארי

בהינתן מערך ממויין, נרצה למצוא איבר מסויים.

נגדיר בצורה רקורסיבית:
$$\begin{cases} T(n) = T\left(\frac{n}{2}\right) + 1 \\ T(1) = 1 \end{cases}$$

נפתור בשיטת האיטרציה:
$$\begin{aligned} T(n) &= T\left(\tfrac{n}{2}\right) + 1 \\ T\left(\tfrac{n}{2}\right) &= T\left(\tfrac{n}{4}\right) + 1 \\ T(n) &= T\left(\tfrac{n}{4}\right) + 2 \\ T\left(\tfrac{n}{4}\right) &= T\left(\tfrac{n}{8}\right) + 1 \\ T(n) &= T\left(\tfrac{n}{8}\right) + 3 \end{aligned}$$

אזי נקבל כי:
$$\begin{aligned} T(n) &= T\left(\tfrac{n}{2}\right) + 1 \\ T(n) &= T\left(\tfrac{n}{4}\right) + 2 \\ &\vdots \\ T(n) &= T\left(\tfrac{n}{2^i}\right) + i \end{aligned}$$

צ"ל באינדוקציה את הניחוש. נגיע לתנאי העצירה כאשר $\frac{n}{2^i} = 1 \Leftrightarrow 2^i = n \Leftrightarrow i = \log n$

כאשר $i = \log n$ אזי $T(n) = T(1) + \log n = \log n + 1$

תרגיל:

$$\begin{cases} T(n) = T(\sqrt{n}) + 1 \\ T(2) = 1 \end{cases}$$
נפתור בעזרת החלפת משתנים: נציב:
$$\begin{aligned} m &= \log n \\ n &= 2^m \\ \sqrt{n} &= 2^{\frac{m}{2}} \end{aligned}$$
,
$$\begin{cases} T(2^m) = T\left(2^{\frac{m}{2}}\right) + 1 \\ T(2) = 1 \end{cases}$$
נגדיר $S(m) = T(2^m)$ ונקבל
$$\begin{cases} S(m) = S\left(\frac{m}{2}\right) + 1 \\ S(1) = 1 \end{cases}$$

אנו יודעים כי $T(2^m) = S(m) = \log m + 1$. נחזור ל־$m = \log n$ ונקבל $T(n) = \log\log n + 1$

שיעור 3

שיטת עצי רקורסיה

(השלמה לשיעור 2)
$$\begin{cases} T(n) = 3T\left(\frac{n}{4}\right) + n^2 \\ T(1) = 1 \end{cases}$$

טענה: ברמה ה־i: 3^i ל־$T\left(\frac{n}{4^i}\right)$

טענה: ברמה ה־i מתבצע: $3^{i-1}\cdot\left(\frac{n}{4^{i-1}}\right)^2$
מספר הרמות בעץ: $i=\log_4 n$

סה"כ:

$$T(n)=\sum_{i=1}^{\log_4 n}3^{i-1}\cdot\left(\frac{n}{4^{i-1}}\right)^2=\sum_{i=0}^{\log_4 n-1}3^i\cdot\left(\frac{n}{4^i}\right)^2=n^2\cdot\sum_{i=0}^{\log_4 n-1}\frac{3^i}{16^i}=$$
$$n^2\cdot\sum_{i=0}^{\log_4 n-1}\left(\frac{3}{16}\right)^i\le n^2\sum_{i=0}^{\infty}\left(\frac{3}{16}\right)^i=n^2\cdot\frac{1}{1-\frac{3}{16}}=\frac{16}{13}\cdot n^2\Longrightarrow T(n)\in\Theta\left(n^2\right)$$

הסתברות

מרחב מדגם: אוסף כל התוצאות האפשריות. נסמן את מרחב המדגם ב־Ω.

פונקציית הסתברות: בהינתן מדגם Ω , פונקציית הסתברות היא פונציה $m:\Omega\longrightarrow[0,1]$ כך ש: $\sum_{\omega\in\Omega}m(\omega)=1$. נסמן את m להיות Pr.

מאורע אלמנטרי: כל איבר $\omega\in\Omega$ נקרא "מאורע אלמנטרי".

מאורע: איחוד של מאורעות אלמנטרים. כלומר, מאורע הוא תת קבוצה של Ω. דוג': אם e הוא מאורע, אזי מתקיים: $Pr(e)=\sum_{\omega\in\Omega}Pr[\omega]$ (סכום המאורעות האלמנטרים)

מאורעות משלימים: יהי A מאורע. נגדיר $\overline{A}=\Omega\backslash A$ ומתקיים $Pr\left[\overline{A}\right]=1-Pr[A]$

איחוד מאורעות זרים: A ו־ B זרים אם $A\cap B=\emptyset$. נקבל: $Pr[A\cup B]=Pr[A]+Pr[B]$

איחוד מאורעות לא זרים: נניח A ו־ B מאורעות לא זרים, $Pr[A\cup B]=Pr[A]+Pr[B]-Pr[A\cap B]$ ומתקיים: $Pr[A\cup B]\le Pr[A]+Pr[B]$

חסם האיחוד: $A_1,\ldots,A_n$ מאורעות, אזי מתקיים: $Pr\left[\bigcup_{i=1}^{n}A_i\right]\le\sum_{i=1}^{n}Pr[A_i]$

שאלה:

נניח כי יש n תאים ו־k כדורים. נזרוק את הכדורים לתאים:

א) מה ההסתברות שכל הכדורים ייפלו בתא הימני ביותר? נסמן את המאורע ב־E_n ומתקיים: $Pr[E_n]=\left(\frac{1}{n}\right)^k$

ב) מה ההסתברות שכל הכדורים ייפלו באותו התא? $Pr\left[\bigcup_{i=1}^{n}E_n\right]=\sum_{i=1}^{n}Pr[E_i]=\sum_{i=1}^{n}\left(\frac{1}{n}\right)^k=\left(\frac{1}{n}\right)^{k-1}$

ג) מהי ההסתברות ששני כדורים נתונים יפלו לאותו תא? $\frac{1}{n}$

ד) מהי ההסתברות ששני כדורים כלשהם יפלו לאותו תא? (כלומר מהי ההסתברות ל־"התנגשות" (collision?

$$Pr[\text{collision}]=Pr\left[\bigcup_{i\neq j}\text{ball i collide with ball j}\right]\le\sum_{i\neq j}Pr[\text{ball i collide with ball j}]=\sum_{i\neq j}\frac{1}{n}=\binom{k}{2}\frac{1}{n}\le\frac{k^2}{2n}$$

משתנה מקרי: בהינתן מרחב מדגם Ω ופונקציית הסתברות Pr, משתנה מקרי x הוא פונקציה $X:\Omega\longrightarrow\mathbb{R}$

תוחלת (expected value): יהי X משתנה מקרי, המקבל ערכים $x_1,x_2,\ldots$ אזי, תוחלת המשתנה x היא: $E[X]=\sum_i Pr[X=x_i]\cdot x_i$ (משקל כל מאורע כפול ההסתברות שהוא יקרה)

לינאריות התוחלת: יהיו X,Y מ"מ (משתנים מקריים) $E[X+Y]=E[X]+E[Y]$

דוגמה:

כמה פעמים נטיל מטבע עד שנקבל עץ ($Tails$)? מרחב המדגם הוא: $\Omega = \begin{matrix} T \\ TH \\ THH \\ \vdots \\ TH\ldots H \end{matrix}$, X יהיה משתנה מקרי המציין את מס' ההטלות עד לקבלת עץ. $\begin{matrix} X=1 \longrightarrow \frac{1}{2} \\ X=2 \longrightarrow \frac{1}{4} \\ \vdots \\ X=i \longrightarrow \frac{1}{2^i} \end{matrix}$

$$E\left[X\right] = \sum_{i=1}^{\infty} Pr\left[X=i\right]\cdot i = \sum_{i=1}^{\infty} \frac{1}{2^i}\cdot i = \frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{16}\cdot 4 + \ldots$$

$$\begin{matrix} = & \frac{1}{2}+ & \frac{1}{4}+ & \frac{1}{8}+ & \frac{1}{16}+ & \ldots & = & 1 \\ & & \frac{1}{4}+ & \frac{1}{8}+ & \frac{1}{16}+ & \ldots & = & \frac{1}{2} \\ & & & \frac{1}{8}+ & \frac{1}{16}+ & \ldots & = & \frac{1}{4} \\ & & & & \frac{1}{16}+ & \ldots & = & \frac{1}{8} \\ & & & & & & & \vdots \end{matrix}$$

ואז טור הסכומים: $1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\ldots = 2$

תרגיל:

נניח מבחן אמריקאי בעל 10 שאלות, לכל שאלה 4 תשובות אפשריות.

א) מה ההסתברות שסטודנט יקבל 100? $Pr\left[\text{Grade=100}\right] = \left(\frac{1}{4}\right)^{10}$

ב) מה ההסתברות שהציון הוא בדיוק 90? $\binom{10}{9}\left(\frac{1}{4}\right)^{9}\cdot\frac{3}{4}$

ג) מה ההסתברות לקבל $10c$ עבור $0 \le c \le 10$? $Pr\left[\text{Grade =10c}\right] = \binom{10}{c}\left(\frac{1}{4}\right)^{c}\left(\frac{3}{4}\right)^{10-c}$

יהי X מ"מ שמציין את הציון. $E\left[X\right] = \sum_{c=0}^{10} Pr\left[\text{Grade=10c}\right]\cdot 10c = \sum_{c=0}^{10}\binom{10}{c}\left(\frac{1}{4}\right)^{c}\left(\frac{3}{4}\right)^{10-c}\cdot(10-c)$

נגדיר X_i להיות הציון בשאלה ה-i($1 \le i \le 10$):

$E\left[X_i\right] = \frac{1}{4}\cdot 10 + \frac{3}{4}\cdot 0 = 2.5$

מתקיים: $X = \sum_{i=1}^{10} X_i$, ואז: $E\left[X\right] = E\left[\sum_{i=1}^{10} X_i\right] \underbrace{=}_{\text{Linear}} \sum_{i=1}^{10}\left[X_i\right] = 10\cdot 2.5 = 25$

שיעור 4

בדיקת חוקיות של סוגריים

1) קיים מס' שווה של סוגריים ימניים וסוגריים שמאליים

2) לכל סוגר שמאלי יש סוגר ימני שמתאים לו

3) לכל רישא של הביטוי אין יותר סוגריים ימניים מסוגריים שמאליים.

מעבר מ-infix ל-postfix:

הביטוי בצורת infix: $2\cdot(2+3)+3\cdot 4-5\cdot 6\cdot(7-4-2)$

הביטוי בצורת postfix: $2,2,3,+,\cdot,3,4,\cdot,+,5,6,\cdot,7,4,-,2,-,\cdot,-$

אלגוריתם:

1) אתחל מחסנית S ריקה

2) לכל סמל c בביטוי משמאל לימין:

2.1) אם c אופרנד (מס' / משתנה) הדפס c

2.2) אחרת c אופרטור

2.2.1) כל עוד S לא ריקה וגם $top\left(S\right)$ קודם (מבחינת סדר פעולות) ל־c אז הדפס את $pop\left(S\right)$

2.2.2) $push\left(S,c\right)$

3) כל עוד S לא ריקה, הדפס $pop\left(S\right)$

שיעור 5

דוגמה ־ מחסנית

$Push\left(x\right)$ ־ הכנס. עלות: $O\left(1\right)$
$Pop\left(x\right)$־ הוצא. עלות: $O\left(1\right)$
$Multi-Pop\left(x\right)$ ־ מרוקן את המחסנית. עלות: $O\left(n\right)$
כאשר n הוא מס' האיברים במחסנית

שאלה מנחה: מהי עלות של סדרה של n פעולות על המחסנית?

ניתוח לשיעורין (Amortized analisis) ניתוח "נחת"

ניתוח לשיעורין זו אסטרטגיה מסויימת לניתוח רצף של פעולות המראה שממוצע העלות לכל פעולה הוא קטן, למרות שישנן פעולות בודדות בתוך הרצף שעלולות להיות יקרות.

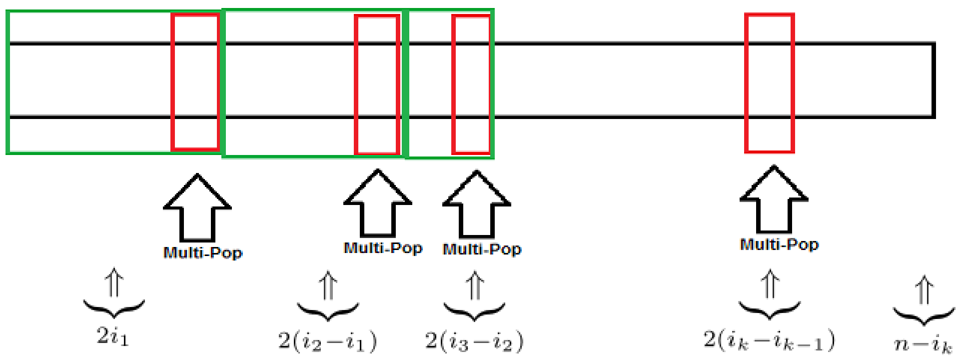
בהינתן רצף פעולות $\sigma_1,\dots,\sigma_n$ ועלויות של הפעולות $c_1,\dots,c_n$, נרצה למצוא חסם הדוק ככל הניתן לסכום העלויות: $T\left(n\right)=\sum_{i=1}^{n}c_i\leq?$

ישנן 3 שיטות:

1. שיטת הצבירה
2. שיטת החיובים (או: "שיטת הבנק")
3. שיטת הפוטנציאל

שיטת הצבירה

נתבונן בסדרה של הכנסות:



$$T\left(n\right)=\sum_{i=1}^{n}c_i\leq 2i_1+2\left(i_2-i_1\right)+2\left(i_3-i_2\right)+\dots 2\left(i_k-i_{k-1}\right)+n-i_k=2i_k+n-i_k=n+i_k\leq 2n$$

שיטת הבנק

בשיטה זו ניתן לכל פעולה עלות שונה מזו שהיא באמת. חלק מהפעולות יקבלו עלויות גבוהות יותר מהעלות האמיתית שלהן. חלק מהפעולות יקבלו עלות נמוכה יותר מהעלות האמיתית שלהן. ברגע שבאתי להכניס איבר אנו נשלם סכום דימיוני בנוסף לסכום האמיתי של העלות שמייצג הוצאה מאוחרת יותר של אותו איבר. העלויות הדמיוניות נשמרות ב־"חיסכון בבנק" שנשתמש בו בשלב מאוחר יותר. את הפעולות הדמיוניות נסמן ב־$\hat{c_i}$.

נרצה להראות שיתקיים: $2n\geq\sum\hat{c_i}\geq\sum c_i$

הגדרנו את $\hat{c_i}$כך: $\hat{c_i}=c_i+\text{Bank deposit}-\text{Bank withdrawal}$

אנו רוצים:

$$2n \geq \sum_{i=1}^{n} \hat{c_i} \geq \sum_{i=1}^{n} c_i$$

ז"א: $\sum_{i=1}^{n} c_i + \text{deposits} - \text{withdrawals} \geq \sum_{i=1}^{n} c_i$ ולמעשה: $\text{deposits} \geq \text{withdrawals}$. אסור להכנס למינוס!
נחזור לדוגמת המחסנית:

$$\hat{c}\,(push) = c\,(push) + \text{deposit} - \text{withdrawal} = 1 + 1 - 0 = 2$$

$$\hat{c}\,(pop) = 1 + 0 - 1 = 0$$

$$\hat{c}\,(Multi - Pop) = k + 0 - k = 0$$

צ"ל כי אף פעם לא נכנס למינוס. קל לראות שבכל שלב היתרה בבנק שווה למספר האיברים כרגע במחסנית. מכאן, לא ניתן להכנס למינוס לעולם וניתן להסיק: $\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c_i} \leq \sum_{i=1}^{n} 2 = 2n$ (החלפנו את $\hat{c_i}$ ב־2. מותר לנו מכיוון שכל פעולה, עלותה היא לכל היותר 2)

שיטת הפונטציאל

נניח שיש לנו סדרת פעולות $\sigma_1, \ldots, \sigma_n$ ונגדיר D_i להיות המצב של מבנה הנתונים לאחר הפעולה ה־i: $D_0 \xrightarrow{\sigma_1} D_1 \xrightarrow{\sigma_2} D_2 \xrightarrow{\sigma_3} \ldots \xrightarrow{\sigma_n} D_n$
בנוסף, נגדיר פונקציית פוטנציאל: $\phi : D \longrightarrow \mathbb{R}$ (D היא קבוצת כל המצבים של מבנה הנתונים)
נגדיר $\hat{c_i}$: $\hat{c_i} = c_i + \phi\,(D_i) - \phi\,(D_{i-1})$ רוצים שיתקיים: $\sum_{i=1}^{n} \hat{c_i} \geq \sum_{i=1}^{n} c_i$

$$\sum_{i=1}^{n} \hat{c_i} = \sum_{i=1}^{n} c_i + \sum_{i=1}^{n} \left[\phi\,(D_i) - \phi\,(D_{i-1})\right] = \sum_{i=1}^{n} c_i + \phi\,(D_n) - \phi\,(D_0)$$

נקבל את המצב שאנו רוצים שיתקיים, כאשר יתקיים: $\phi\,(D_n) \geq \phi\,(D_0)$
בד"כ בוחרים פונקציה בצורה הבאה: אם במצב D_i הפעולה הבאה היא הפעולה הכבדה, אזי $\phi\,(D_i)$ יהיה העלות של אותה פעולה כבדה.
פונקציית הפוטנציאל אם כן במקרה של המחסנית תחזיר את מספר האיברים כרגע במחסנית.

$$\hat{c}\,(push) = c\,(push) + \phi\,(D_i) - \phi\,(D_{i-1}) = 1 + (k+1) - k = 2$$

$$\hat{c}\,(pop) = c\,(pop) + \phi\,(D_i) - \phi\,(D_{i-1}) = 1 + (k-1) - k = 0$$

$$\hat{c}\,(Multi - Pop) = c\,(Multi - Pop) + \phi\,(D_i) - \phi\,(D_{i-1}) = k + 0 - k = 0$$

קל לראות שמתקיים לכל i: $\phi\,(D_i) \geq \phi\,(D_0) = 0$ ולכן: $\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c_i} \leq 2n$

דוגמה נוספת ־ מונה בינארי

נגדיר kביטים המקבלים ערך 0 או 1 , ונגדיר פעולה הנקראת $Increment$ המעלה את ערך הסיביות באחד:

$$\begin{array}{rl} 0 = & \overbrace{0000000}^{\text{k bits}} \\ & \qquad\quad +1 \\ 1 = & 0000001 \\ & \qquad\quad +1 \\ 2 = & 0000010 \\ & \qquad\quad +1 \\ 3 = & 0000011 \end{array}$$

מהי עלות סדרת n פעולות $Increment$?

שיטת הצבירה

$T\,(n) = \sum_{i=1}^{k} c_i =$סה"כ הביטים שהוחלפו$= \sum_{i=1}^{k}$ מספר הפעמים שהביט ה־i התחלף $= \sum_{i=1}^{k} \frac{n}{2^{i-1}} = n \cdot \sum_{i=0}^{k-1} \frac{1}{2^i} \leq 2n$

שיטת הפוטנציאל

נרצה למצוא פונקציית פוטנציאל. הפונקציה תהיה מספר האחדות במונה. קל לראות שמתקיים: $\phi\,(D_i) \geq \phi\,(D_0)$.

נניח כי לפני הפעולה ־ סה"כ k אחדות, ו־t אחדות בזנב. $10001\ldots0\overbrace{11111}^{t}$
$10001\ldots100000$

ולכן:

$$\hat{c}\,(Increment) = c\,(Increment) + \phi\,(D_i) - \phi\,(D_{i-1}) = (t+1) + k - t + 1 - k = 2$$

מכאן: $\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c_i} \leq 2n$

דוגמה נוספת: מערך דינאמי

נניח כי יש לנו מערך בגודל n ואנו מכניסים אליו איברים. אם המערך מלא, נגדיר מערך חדש גדול פי 2 , נכניס את האיבר החדש למקום ה:$n+1$.

מהי העלות של סדרת m הכנסות?

$O\left(m\cdot n\right)$ (לא הדוק!) , ננסה "להדק" את החסם מלעיל בעזרת:

שיטת הבנק

נשלם 3 לכל איבר: 1 עבור ההכנסה עצמה, 1 עבור ההעתקה שלו הלאה ו־1 עבור ההעתקה של "חבר".

שיטת הפוטנציאל

נגדיר 2 משתנים: גודל נוכחי ־ $size$ומספר האיברים כרגע ־ num. אחרי ההעתקה: $size=2\cdot num$רגע לפני ההעתקה: $size=num$
לפיכך פונקציית הפוטנציאל תהיה: $\phi\left(D_i\right)=2num_i-size_i$

שיעור 6

ייצוגים של גרפים

$G=(V,E)$
$|V|$ מספר הקודקודים בגרף
$|E|$ מספר הקשתות בגרף

מטריצת שכנויות Adjacency Matrix

מניחים $V=\{1,2,\ldots,|V|\}$ ואז $A=\{a_{ij}\}$ תהיה המטריצה:

$$\begin{matrix}0&0&1&1&1&1\\0&0&0&0&0&0\\1&0&0&1&0&0\\1&0&1&0&0&1\\1&0&0&0&0&0\\1&0&0&1&0&0\end{matrix}$$

כאשר

$$a_{ij}=\begin{cases}1 & (i,j)\in E\\0 & \text{else}\end{cases}$$

רשימת שכנויות: נרשום את כל הקודקודים (למשל במערך), ועבור כל אחד, נחזיק רשימה מקושרת של שכניו בצורה שרירותית

	מטריצת שכנויות:	רשימת שכנויות:
גודל הייצוג	$O\left(\|V\|^2\right)$	$O\left(\|V\|+\|E\|\right)$
זמן בדיקה $(u,v)\in E$	$O\left(1\right)$	$O\left(\deg\left(u\right)\right)$
זמן חישוב דרגה של u	$O\left(\|V\|\right)$	$O\left(\deg\left(u\right)\right)$

חיפוש לרוחב BFS - Breadth First Search

קלט: גרף $G=(V,E)$ וקודקוד $s\in V$.

מבני נתונים: לכל קודקוד נשמור את צבעו:
color$[u]$ לבן־ עוד לא ביקרנו, אפור ־ ביקרנו אבל עוד לא סיימנו, שחור ־ סיימנו.
d$[u]$ ־ בסיום ריצת BFS על s $\delta\left(s,u\right)$ $d\left[u\right]$ $\forall u\in V$ כאשר $\delta\left(s,u\right)$ הינו אורך המסלול הקצר ביותר מ־s ל־u.
תור Q של קודקודים.
$\prod\left[u\right]$ יהיה "מסלול אחורה" עבור כל קודקוד, מי הגיע אליו ראשון.

BFS(G,s)

1. for each a∈V$[G]\setminus s$
   (a) color$[u]\longleftarrow white, d\left[u\right]\longleftarrow\infty,\prod\left[u\right]\longleftarrow Null$
2. color$[s]\longleftarrow gray, d\left[s\right]\longleftarrow 0,\prod\left[s\right]\longleftarrow Null$
3. Q$\longleftarrow\{s\}$
4. while Q is not empty
   (a) u$\longleftarrow$Q.dequeue()
   (b) for each v∈ $adj\left[u\right]$

i. if color$[v]=white$
   A. color$[v]\longleftarrow gray$
   B. d$[v]$⟵d$[u]$+1
   C. $[v]\longleftarrow d\,[u]+1$
   D. $.\prod[v]\longleftarrow u$
   E. Q.enqueue(v)

ii. color$[u]\longleftarrow black$

חיפוש לעומק DFS -Depth First Search

קלט: גרף $G=(V,E)$ וקודקוד $s\in V$.

מבני נתונים: לכל קודקוד נשמור את צבעו:
color$[u]$ לבן־ עוד לא ביקרנו, אפור ־ ביקרנו אבל עוד לא סיימנו, שחור ־ סיימנו.
d$[u]$ ־זמן הכניסה אל u,
f$[u]$ ־ זמן היציאה מ־u.
$\prod[u]$

DFS(G)

1. for each u∈V$[G]$
   (a) color$[u]\longleftarrow white$,$\prod[u]$⟵ $Null$
2. time⟵0
3. for each u∈ $V\,[G]$
   (a) if color$[u]$ = white
      i. DFS_VISIT(G,u)

DFS_VISIT(G,u)

1. color$[u]$⟵gray
2. d$[u]$⟵ $time\longleftarrow time+1$
3. for each v∈ $adj\,[u]$
   (a) if color$[u]$ = white
      i. $\prod(v)\longleftarrow u$
      ii. DFS_VISIT(G,v)
   (b) f$[u]\longleftarrow time\longleftarrow time+1$
   (c) color$[u]\longleftarrow black$

סיווג קשתות:

קשתות עץ: $\forall u\in V\left(\prod(u),u\right)$

קשת אחורה: קשת מצאצא לאב קדמון

קשת קדימה: קשת מאב קדמון לצאצא לא ישיר

קשת חוצה: קשת מ־u ל־v כאשר u אינו צאצא ולא אב קדמון של v.

משפט הסוגריים (הבערך):

- u ו־v אינם אב קדמון אחד של השני אסס $[d[u], f[u]] \cap [d[v], f[v]] = \emptyset$
- u אב קדמון של vאסס $d[u] \le d[v] \le f[v] \le f[u]$

זמני הריצה של DFS ו־BFS הם $O(|V| + |E|)$

שיעור 7

מיון טופולוגי

קלט:

גרף מכוון $G = (V, E)$ ללא מעגלים.

פלט: סדר על הקודקודים V כך שלכל $(u, v) \in F$ מתקיים ש־u מופיע לפני v בסדר.

אלגוריתם: הרץ $DFS(G)$, וכל פעם שקודקוד נצבע בשחור, נוסיף אותו לתחילתה של רשימה מקושרת L. לבסוף נחזיר את L. (סדר יורד של זמני סיום)

זמן: $O(|V| + |E|)$

$B - tree$

מודל הזיכרון החיצוני. העלות: מס' הגישות לזכרון החיצוני. (עבודה ב־RAM היא בחינם לפי המודל).
פרמטר $O(m) \approx$ גודלו של בלוק מידע.

הגדרה: עץ־B עבור פרמטר mהינו עץ חיפוש $m - ary$ (בעל m בנים בערך) כך ש:

1. לכל קודקוד חוץ מהשורש יש בין $\lceil \frac{m}{2} \rceil$ לבין m בנים.
2. לשורש יש בין 2 לבין m בנים.
3. לקודקוד עם k בנים יש $k - 1$ מפתחות (איברים).
4. כל העלים באותה הרמה, ואינם מכילים מידע $(Null)$

מהו הגובה של $B - tree$?

נסתכל על עץ עם מס' מינימלי של איברים:

רמה	מס' קודקודים	סה"כ מס' איברים
0	1	1
1	2	m
2	m	$m \cdot \frac{m}{2}$
3	$\frac{m^2}{2}$	$\frac{m^2}{2} \cdot \frac{m}{2}$
l	$\frac{m^{l-1}}{2^{l-2}}$	$\frac{m^l}{2^{l-1}}$

סה"כ: $O\left(\left(\frac{m}{2}\right)^l\right)$

$$\left(\frac{m}{2}\right)^l \approxeq N$$
$$l \cdot \log \frac{m}{2} \approxeq \log N$$
$$l \approxeq \frac{\log N}{\log m - 1} = O(\log_m N)$$

חיפוש: בדומה לעץ חיפוש בינארי, $O(\log_m N)$

הכנסה: מתבצעת הכנסה אם חוקית. אם לא, אז מתבצע פיצול, והאמצע עולה רמה. אם הוא גורם לבעיה, גם שם מתבצע פיצול והאמצע עולה רמה, זמן $O(\log_m N)$

מחיקה: נשים לב שמספיק לדעת איך למחוק עלה. נחליף את הערך של העלה באיבר הבא אחריו (ימינה בעץ ושמאלה עד שמגיעים לאחרון) זה לא מספיק טוב, ויש הסבר על כך בתרגול. (הזמן יהיה $O(\log_m N)$)

עץ 3־2 :

זהו $B - tree$ עם $m = 3$.

B^*/B^+:

כל המידע נמצא בעלים.

שיעור 8

תרגול על עצים

הגדרה: משפחה של עצים תקרא "מאוזנת" אם כל עץ השייך למשפחה הוא בעל גובה $O(\log n)$ כאשר n הוא מס' הצמתים בעץ.

תרגיל:

לכל תכונה בהמשך, קבע האם משפחה של עצים המקיימת את התכונה גורר שהמשפחה מאוזנת. אם כן - הוכח, אם לא - הצג דוגמה נגדית.

א) כל צומת בעץ הוא או עלה או שיש לו שני בנים.

פתרון: לא מאוזן. דוגמה נגדית:

/Tree [.x [.x [.x [.x [.... [.x [.x]]]]]]]

ב) הגודל של כל תת־עץ יכול להכתב כ־ 2^k-1 , כאשר k הוא מספר שלם (k איננו זהה לכל תת עץ)

פתרון: מאוזן.

הוכחה: נתבונן בתת עץ כלשהו עם שורש r, אנו יודעים שגודלו הוא $2^{k_1}-1$, גודל תת העץ השמאלי $2^{k_2}-1$ וגודל תת העץ הימני הוא $2^{k_3}-1$. נקבל כי: $|T(r)| = |T(left(r))| + |T(right(r))| + 1$ אזי מתקיים:

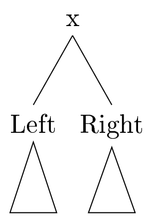
$$\begin{gathered}2^{k_1}-1 = 2^{k_2}-1+2^{k_3}-1+1\\ 2^{k_1} = 2^{k_2}+2^{k_3}\end{gathered}$$

$k_1 \ge k_2$ וגם $k_1 \ge k_3$ נניח בה"כ $k_3 \ge k_2$ אזי $2^{k_1-k_2} = 1+2^{k_3-k_2}$ ונקבל $2^1 = 1+2^0$, כלומר $k_2 = k_3$ ו־ $\begin{matrix}k_1-k_2=1\\ k_1=k_2+1\end{matrix}$ ולכן העץ בהכרח מאוזן לחלוטין (הוא מלא).

ג) קיים קבוע $0 < c < 1$ כך שלכל צומת בעץ, גודל תת העץ הקטן יותר הוא לפחות c פעמים גודל תת העץ הגדול יותר. $c\cdot|T_1| \le |T_2| \le |T_1|$

פתרון: מאוזן.

הוכחה: באינדוקציה על n נניח שהטענה נכונה לכל $k < n$ כלומר, כל עץ בעל k קודקודים, גובהו חסום ע"י $d\cdot\log k$. נתבונן בעץ בעל n קודקודים כשבתת העץ השמאלי יש n_1 קודקודים ובימני n_2 קודקודים. (מתקיים $n = n_1+n_2+1$)



גובה תת העץ השמאלי חסום ע"י $d\log n_1$ וגובה תת העץ הימני חסום ע"י $d\log n_2$ אזי גובה העץ הוא $h = max\{d\log n_1, d\log n_2\}+1$.

נניח בה"כ $n_1 \ge n_2$ אזי לפי התכונה יתקיים: $n_1 \ge n_2 \ge c\cdot n_1$. אנו יודעים כי מתקיים
$$\begin{aligned}n = n_1+n_2+1 \quad &\ge n_1+c\cdot n_1+1\\ &\ge (c+1)\cdot n_1+1\\ &\ge (c+1)\cdot n_1\end{aligned}$$
מכאן, $n_1 \le \frac{1}{c+1}\cdot n$.

נציב ב־ $d\log n_1+1$ את מה שמצאנו ונקבל: $d\log n_1+1 \le d\log\left(\frac{1}{c+1}\cdot n\right)+1 = d\log n - d\log(c+1)+1 \underline{\le d\log n}$ וזה (המסומן בקו תחתון)

מתקיים כאשר $\begin{matrix}d\log(c+1)\ge 1\\ d\ge\frac{1}{\log(c+1)}\end{matrix}$.

ד) קיים קבוע c (שלם) כך שלכל צומת של העץ הגבהים שונים בלכל היותר c.

תשובה: מאוזן. ההוכחה באתר הקורס.

ה) הגובה הממוצע של כל קודקוד הוא $O(\log n)$

פתרון: לא מאוזן. ניקח עץ מלא, ונאריך אותו עם "עץ שרוך" בקודקוד עלה כלשהו בתחתית. נניח עץ שבו יש $n-\sqrt{n}$ צמתים הם ממש עץ מלא מאוזן, ושאר $\sqrt{n}$ הצמתים הם מעין שרוך.
אזי גובה העץ ללא השרוך הוא $\log(n-\sqrt{n}) \le \log(n)$, גובה השרוך הוא $\sqrt{n}$ ולכן גובה העץ כולו יהיה $\Omega(\sqrt{n})$. נראה שהגובה הממוצע מקיים את התנאי:
גובה$\le$ $\frac{1}{n}\left(\sqrt{n}\cdot(\sqrt{n}+\log n)+(n-\sqrt{n})\cdot\log n\right) = \frac{1}{n}\left(n+\sqrt{n}\log n+n\log n-\sqrt{n}\log n\right) = \frac{1}{n}(n+n\log n) = \log n+1$
ממוצע, והתנאי מתקיים.

תרגיל: עץ שלם הוא עץ שבו לכל קודקוד שני בנים או שהוא עלה. הוכח באינדוקציה על מבנה העץ שלעץ שלם בעל n עלים יש $n-1$ קודקודים פנימיים.

הוכחה:

בסיס: עץ עם עלה יחיד יש לו 0 קודקודים פנימיים.

צעד: נניח שעבור על העצים השלמים בעלי $k \le n$ עלים יש $k-1$ קודקודים פנימיים. נוכיח עבור העצים השלמים בעלי n עלים.
יהי עץ שלם בעל n עלים. מכיוון שהוא עץ שלם, לכל קודקוד פנימי שני בנים, בפרט לשורש. נתבונן בשורש, תת העץ הימני הוא עץ שלם בעל n_1 עלים ולכן הוא מכיל n_1-1 קודקודים פנימיים. תת העץ השמאלי הוא שלם ויש לו $n-n_1$ עלים, ולכן $n-n_1-1$ קודקודים פנימיים.
נסכם את סך הקודקודים הפנימיים בעץ: $1+(n_1-1)+(n-n_1-1)=n-1$, והוכחנו את הטענה.

תרגיל: נתונים 2 עצי חיפוש בינאריים. הצג אלגוריתם הממזג אותם לעץ חיפוש מאוזן ממוזג.

פתרון: נבצע מעבר $in-Order$ על T_1 ו־T_2 ונקבל 2 רשימות ממוינות. נמזג לרשימה ממויינת. ניקח את החציון להיות השורש, כל מה שקטן ממנו להיות תת העץ השמאלי וכל מה שגדול ממנו להיות תת העץ הימני. נציג פסאודו קוד:

```
generateBinaryTree(A,i,j)
        if j=i+1
                return makeLeaf(A[i])
        else
                median=(i+j/2)
                T1 <- generateBinaryTree(A,i,med-1)
                T2 <- generateBinaryTree(A,med+1,j)
                return tree(T1 , med , T2)
```

תרגיל: בנה מבנה נתונים התומך בפעולות הבאות:

1. הכנסה
2. הוצאה
3. חיפוש
4. מציאת האיבר ה־k בגודלו (כלומר, $k-1$ מהאיברים קטנים ממנו, והשאר גדולים ממנו)

הפתרון יהיה בעזרת עץ. על מנת שמציאת האיבר ה־k בגודלו יקח לנו $O(\log n)$
אלגוריתם:

```
OS-Select(X,i)   // X is a node (or the root) and we're looking for the i'th number ascending
        r <- size[left(x)] + 1
        if r = i
                return x
        else
                if i < r then
                        OS-Select([left[x],i)
                else
                        return OS-Select(right[x],i-r)
```

שיעור 9

להשלים

שיעור 10

בעיה: נתונה קבוצה של מספרים $u = \{0, \ldots, u-1\}$ ותת־קבוצה $S \subset U$, ונניח ש־$|U| << |S|$ ו־ $|S| = n$ [1]
נרצה להיות מסוגלים לענות על השאילתא הבאה:
$query\,(x)$ ־ האם $x \in S$, אם יש מידע, החזר.

לדוגמה

S קבוצת סטודנטים ־ מפתח ־ ת"ז $|S| = 20$ (יש 20 סטודנטים)

U קבוצה של כל ת"ז האפשריים $|U| \approx 10^9$

פתרון ראשון: לא מניחים כלום על S (מתעלמים מהעובדה ש־$S \subset U$) שומרים רשימה/מערך ממויין, עץ חיפוש מתאים. אזי חיפוש לפחות $O\,(\log n)$.

פתרון שני: מערך בגודל U.
חיפוש: $O\,(1)$, בניה: $O\,(U)$, מקום: $O\,(U)$

פתרון שלישי: $HashMap$. שומרים מערך בגודל m, כאשר $m \geq n$, ופונקציית ה־$Hash$ היא $h : U \longrightarrow \mathbb{Z}_m$. היינו רוצים m לינארי ב־n. שומרים את h.

מה זאת אומרת "שומרים את h "? יש $m^{|u|}$ פונקציות.
אם ניקח את הפונקציה הממפה ערכי מפתחות לאיברים, פונקציה כללית כזו תיקח $O\,(|U|)$, כי נזדקק להגדיר לכל $u \in U$ לאיזה ערך $x \in [0, m-1]$ היא ממפה אותו.

מטרות:

1) ה־"מיפוי" וה־"פונקציה" h , צריכה להיות בעלת תיאור קומפקטי.

2) היינו רוצים ש־m יהיה לינארי ב־n.

3) רוצים שביחס ל־S ל־h אין התנגשויות בכלל.

הגדרה: בהינתן קבוצה $S \subset U$ נאמר שהפונקציה $h : U \longrightarrow \mathbb{Z}_m$ היא פונקציית "ערבוב" ("גיבוב") מושלמת אם לכל $x, y \in S$ אם $x \neq y$ אזי מתקיים $h\,(x) \neq h\,(y)$.[2]
נרצה למצוא אלגוריתם למציאת פונקציה h העונה על התנאים הנ"ל.

תכונה של פונקציה אקראית

נניח כי בחרנו באקראי פונקציה h מכלל הפונקציות שממפות את U לתוך $\mathbb{Z}_m$. לכל $x, y \in U$, כאשר $x \neq y$
מה ההסתברות ש־$h\,(x) = h\,(y)$? $Pr\,[h\,(x) = h\,(y)] = \frac{1}{m}$

פונקציית גיבוב אוניברסלית Universal Hash

הגדרה: משפחת פונקציות $Hash$, $H = \{h\,|\,U \longrightarrow \mathbb{Z}_m\}$ תיקרא "משפחה אוניברסלית" אם לכל $x, y \in U$ כאשר $x \neq y$, מתקיים $Pr_{h \in H}\,[h\,(x) = h\,(y)] \leq \frac{1}{m}$.

טענה: קיימות משפחות אוניברסליות עם ייצוג קומפקטי.

[1] $<<$ זה סימון לקטן בסדר גודל משמעותי
[2] חח"ע ביחס ל־S, ולא ביחס ל־U!

דוגמה

$H_{p,m}=\left\{h_{a,b}\,\middle|\,\begin{matrix}1\le a\le p-1\\0\le b\le p-1\end{matrix}\right\}$ [3] כאשר $h_{a,b}=((ax+b)\mod p)\mod m$ ו־p הוא ראשוני גדול מ־m.

יש בערך p^2 פונקציות במשפחה.

בניית פונקציה Perfect Hash עבור הקבוצה S.

קלט: הקבוצה $S\subset U$.

פלט: פונקציה $h:U\longrightarrow\mathbb{Z}_m$, כך שלכל $x,y\in S$, אם $x\neq y$ אז $h(x)\neq h(y)$

תיאור האלגוריתם:

1. נקבע משפחה אוניברסלית H.
2. נבחר פונקציה h מתוך H באקראי.
3. נבדוק אם h מושלמת ־ ניצור קבוצות $S_i=\{x\in S\,|\,h(x)=i\}$.
4. אם קיים i כך ש־$|S_i|>1$ נחזור לשלב 2.

ניתוח האלגוריתם:

$$Pr_{h\in H}\left[\text{There's a collision}\right]=Pr\left[\bigvee_{\substack{x,y\in S\\x\neq y}}h(x)=h(y)\right]\le\sum_{\substack{x,y\in S\\x\neq y}}Pr\left[h(x)=h(y)\right]$$

$$\underbrace{\le}_{\text{h is universal}}\sum_{\substack{x,y\in S\\x\neq y}}\frac{1}{m}=\binom{|s|}{2}\cdot\frac{1}{m}\underbrace{=}_{|s|=n}\frac{n(n-1)}{2m}\le\frac{n^2}{2m}$$

אם נבחר $m=n^2$, אזי $Pr_{h\in H}\left[\text{There's a collision}\right]\le\frac{1}{2}$ בתוחלת, נבדוק 2 פונקציות עד שנמצא פונקציה מושלמת.

סיכום: מצאנו מבנה נתונים ללא התנגשויות בכלל, בגודל $\approx n^2$. עלות בתוחלת, של הבנייה: $O\left(n^2\right)=O(m)$, עלות חיפוש תהיה $O(1)$.
כמו כן, הייצוג של h קומפקטי.

ראינו: $Pr_{h\in H}\left[\text{There's a collision}\right]\le\frac{n^2}{2m}$, אפשר להראות כי: $Pr_{h\in H}\left[\text{number of collisions }\ge k\right]\le\frac{n^2}{2mk}$.
בפרט, אם $m=n$, $k=n$, אזי: $Pr_{h\in H}\left[\text{number of collisions}\ge n\right]\le\frac{1}{2}$ (מכאן אפשר להסיק $Pr_{h\in H}\left[\text{number of collisions}<n\right]\ge\frac{1}{2}$)

אזי אם $\binom{k}{2}\approx n$ אזי $\frac{k(k-1)}{2}\approx n$, ומתקיים $k\approx\sqrt{n}$ (והבנייה תהיה $O(n)=O\left(\left(\sqrt{n}\right)^2\right)$)

אלגוריתם FKS

1. נקבע משפחה אוניברסלית H (התלויה ב־m כאשר $m=n$)
2. נבחר פונקציה $h\in H$ באקראי.
3. נבדוק כמה התנגשויות יש לנו: ניצור קבוצות $S_i=\{x\in S\,|\,h(x)=i\}$
4. אם מספר ההתנגשויות הכולל $\le n$, אז חזור לשלב 2.
5. לכל i, אם $|S_i|\le 1$ ־ לא עושים כלום. אחרת:
   (א) נקבע משפחה $H_{\left|S_i^2\right|}$
   (ב) נבחר באקראי h מתוך H
   (ג) אם יש התנגשות, נחזור ל־(a)

[3] הוכחה מדוע זוהי משפחה אוניברסלית ־ בספר של Corman

ניתוח האלגוריתם: מספר הפעמים שנבצע את שלבים 2 עד 4 , בתוחלת: 2. עלות כל סיבוב: $O\left(n\right)$ לכן עלות עד שנגיע לשלב 5 , בתוחלת, $O\left(n\right)$.

כמה עולה סה"כ שלב 5?

עבור i ספציפי: $O\left(\left|S_i\right|^2\right)$ (בתוחלת) ולכן בסה"כ עלות שלב 5, היא: $\text{Total Cost} \leq \sum\limits_{i=1}^{n}\left|S_i\right|^2 \leq 4\sum\limits_{i=1}^{n}\binom{\left|S_i\right|}{2} < 4n$

$$\left|S_i\right|^2 \leq 4\cdot\binom{\left|S_i\right|}{2} = \frac{4\left|S_i\right|\left(\left|S_i\right|-1\right)}{2}$$
$$\left|S_i\right|^2 \leq 2\left|S_i\right|^2 - 2\left|S_i\right|$$
$$2 \leq \left|S_i\right|^2$$

עלות האלגוריתם בשלב 5 היא $O\left(n\right)$ ולכן עלות האלגוריתם בסה"כ הוא $O\left(n\right)$ בתוחלת.

מקום: מערך בגודל $O\left(n\right)$.

$\text{Total size of mini-arrays} \longrightarrow \sum\limits_{i=1}^{n}\left|S_i\right|^2 < 4n$

אמרנו: $Pr_{h\in H}\left[\text{number of collisions } \geq k\right] \leq \frac{n^2}{2mk}$

אפשר להראות כי תוחלת מספר ההתנגשויות: $Pr_{h\in H}\left[\text{number of collisions }\right] \leq \frac{n^2}{2m}$ (כמעט אותו חישוב כמו ההסתברות שקיימת התנגשות).

X - מספר ההתנגשויות.

$X_{i,j}$ - משתנה מקרי המקבל: 1 אם i,j מתנגשים , 0 אחרת.

$$\mathbb{E}\left[X_{i,j}\right] = 1\cdot Pr\left[X_{i,j}=1\right] + 0\cdot\left[X_{i,j}=0\right] \leq \frac{1}{m}$$

ולכן: $X = \sum\limits_{\substack{i,j\in S\\ i\neq j}} X_{i,j}$

$$\mathbb{E}\left[X\right] = \mathbb{E}\left[\sum_{\substack{i,j\in S\\ i\neq j}} X_{i,j}\right] = \sum_{\substack{i,j\in S\\ i\neq j}}\mathbb{E}\left[X_{i,j}\right] \leq \binom{n}{2}\cdot\frac{1}{m} \leq \frac{n^2}{2m}$$

אי שוויון מרקוב

יהי X מ"מ אי שלילי , ויהי $k>0$ אזי: $Pr\left[X\geq k\right] \leq \frac{\mathbb{E}[x]}{k}$

הוכחה:

$$\mathbb{E}\left[x\right] = \sum_{x} Pr\left[X=x\right] = \sum_{x<k} x\cdot Pr\left[X=x\right] + \sum_{x\geq k} x\cdot Pr\left[X=x\right] \geq \sum_{x\geq k} x\cdot Pr\left[X=x\right] \geq \sum_{x\geq k} k\cdot Pr\left[X=x\right] \geq k\sum_{x\geq k} Pr\left[X=x\right] \geq kPr\left[X$$

לכן:

$Pr\left[\text{number of collisions } \geq k\right] \leq \frac{\mathbb{E}[x]}{k} \leq \frac{n^2}{2m\cdot k}$

שיעור 11

אלגוריתם Select

נזכיר את אלגוריתם QuickSort: בחירת הציר ,Pivot, ומיון האיברים לקטנים ממנו וגדולים ממנו, מימינו ומשמאלו.

עלות QuickSort נחלקת למקרים לפי בחירת הציר:

במקרה שבחרנו את הציר להיות החציון, נקבל את נוסחת הרקורסיה $T\left(n\right) = 2T\left(\frac{n}{2}\right) + O\left(n\right)$ שכבר ראינו שהיא רצה ב־$O\left(n\log n\right)$.

גם אם לא בחרנו את החציון, אלא בחרנו מיפוי αn שמאלה ו־$\left(1-\alpha\right)n$ ימינה , עבור $0<\alpha<1$, נקבל את נוסחת הרקורסיה $T\left(n\right) = T\left(\alpha n\right) + T\left(\left(1-\alpha\right)n\right) + O\left(n\right)$, וגם אותה ראינו שהיא רצה ב־$O\left(n\log n\right)$.

אם ניקח Pivot"רע" , (נניח כל פעם ניקח את האיבר החמישי בגודלו להיות הציר). נקבל $T\left(n\right) = T\left(k\right) + T\left(n-k\right) + O\left(n\right)$ וזמן ריצה של $O\left(n^2\right)$.

אלגוריתם Select:

קלט:

- מערך לא ממויין A בגודל n
- מספר $1 \leq k \leq n$

פלט: האיבר ה־k בגודלו במערך.

רעיון: נבחר פיבוט, ונבדוק רק בצד ה"נכון" כל פעם.
בחירת Pivot טוב: $T(n) = T(\alpha n) + O(n) \in O(n)$, בחירת Pivot רע: $T(n) = T(n-c) + O(n) \in O(n^2)$

נראה אלגוריתם:

```
Randomized-Select(A,P,r,i) //search the i-th sized item in sub array A[p...r]
    if p=r
        return A[p]
    q <- Randomized-Partition(A,p,r) //choose a random pivot, then perform partition
                                       and return the pivot's index.
    k <- (q-p-1)
    if (i <= k)
        return Randomized-Select(A,P,q,i)
    else
        return Randomized-Select(A,q+1,r,i-k)
```

ניתן להראות שתוחלת זמן הריצה הוא $O(n)$.

פתרון בעיית הבחירה (אלגוריתם Select) בזמן לינארי במקרה הגרוע

1. אם $n=1$ אז Select יחזיר את איבר הקלט היחיד, כאיבר ה־i הקטן ביותר
2. אחרת, (אם $n>1$)
   (א) חלק את n האיברים במערך הקלט ל־$\frac{n}{5}$ קבוצות בנות 5 איברים בכל קבוצה
   (ב) מצא את החציון של כל אחת מ־$\frac{n}{5}$ הקבוצות
   (ג) הפעל את Select באופן רקורסיבי כדי למצוא את החציון x של $\frac{n}{5}$ החציונים שנמצאו ב־(b)
   (ד) חלק את מערך הקלט סביב חציון החציונים x (Partition) וקבל חזרה אינדקס k , האינדקס של הפיבוט לאחר ה־Partition
   (ה) הפעל רקורסיבית את האלגוריתם:
      i. אם $i \leq k$: על צד שמאל.
      ii. אם $i > k$: על צד ימין עם $i-k$.

ניתוח האלגוריתם: $(a)+(b)$ יחדיו עולים $O(n)$. $(d)+(e)$ בדיוק כמו ב־Randomized-Select.

הטענה: חציון החציונים שמצאנו ב־(c) הוא Pivot "טוב" . מכמה איברים הוא גדול בוודאות? גדול מחצי מהחציונים כלומר: $\frac{1}{2} \cdot \frac{n}{5} = \frac{n}{10}$. כל חציון כזה גדול מעוד 2 בחמישייה שלו, ולכן חציון החציונים גדול בסה"כ מ־$\frac{3n}{10}$ מהאיברים. בצורה סימטרית, חציון החציונים קטן בוודאות מ־$\frac{3n}{10}$ איברים. מכאן, במקרה הגרוע ביותר, נרוץ רקורסיבית על $\frac{7n}{10} = 70\%$ מהאיברים.
סה"כ: $T(n) \leq \underbrace{T\left(\frac{n}{5}\right)}_{(c)} + \underbrace{T\left(\frac{7n}{10}\right)}_{(d)+(e)} + \underbrace{O(n)}_{(a)+(b)}$

טענה: $T(n) \in O(n)$. כלומר, קיים קבוע c כך ש: $T(n) \leq c \cdot n$.

הוכחה באינדוקציה: בסיס $n=1$, טריוויאלי.
לכל $T(n)$

צעד: $T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + d \cdot n$ עבור קבוע $d > 0$.
$\leq c \cdot \frac{n}{5} + c \cdot \frac{7n}{10} + d \cdot n = \frac{9n}{10} \cdot c + d \cdot n \underbrace{\leq}_{\text{needs proving}} c \cdot n$
$dn \leq \frac{1}{10} c \cdot n$
$10d \leq c$

מה היה קורה לו היינו מחלקים לשלישיות במקום לחמישיות?

מכמה איברים גדול בוודאות חציון החציונים? גדול מחצי מהחציונים: $\frac{1}{2}\cdot\frac{n}{3}=\frac{n}{6}$, כל אחד כזה גדול מעוד אחד , סה"כ $\frac{2n}{6}=\frac{n}{3}$.
נקבל נוסחה $T\left(n\right)\leq T\left(\frac{n}{3}\right)+T\left(\frac{2n}{3}\right)+O\left(n\right)$, ועלותה הוא $O\left(n\log n\right)$

תרגיל

איבר במערך מגודל n ייקרא "דומיננטי" אם הוא מופיע לפחות $\frac{n}{2}$ פעמים. הצע אלגוריתם שבהינתן מערך A בגודל n מוצא אם קיים ב־A איבר דומיננטי, ואם יש, מהו.

פתרון: האיבר הדומיננטי הוא בהכרח החציון. (למה?) נריץ את אלגוריתם Select עם $i=\frac{n}{2}$ ונקבל חזרה את החציון. נרוץ על המערך, ונספור כמה פעמים הוא מופיע: אם $\frac{n}{2}$ לפחות, נחזיר אותו, אחרת נחזיר תשובה שלילית.

תרגיל

בהינתן מערך A של n מספרים ומספר טבעי $k<n$, ברצוננו למצוא את כל האיברים ה־$i\cdot k$־ים בגודלם עבור $i=1,\ldots,\frac{n}{k}$.
תארו אלגוריתם הרץ בזמן $O\left(n\log\frac{n}{k}\right)$ שמוצא את כל האיברים הללו. (ניתן להניח ש־n ו־ k חזקות של 2)

פתרון: נמצא את החציון, ונריץ את האלגוריתם שוב על שני הצדדים.