פתרון תרגיל מספר 10 ־ דאסט

שם: מיכאל גרינבאום, **ת"ז:** 211747639

5 ביוני 2019

1. **פתרון:**

(א) **צ"ל:** נוכיח שהאלגוריתם הוא שקול ל$\frac{1}{2}i+\frac{1}{2}i^2$

הוכחה:

נוכיח באינדוקציה על אינדקס הריצה i, ונגדיר j_k להיות הj המתקבל בריצה הj,

<u>בסיס:</u> $i=0$, נשים לב כי

$$j_0=j=j+\frac{1}{2}\cdot 0+\frac{1}{2}\cdot 0^2$$

כלומר הטענה נכונה עבור $i=0$,

<u>צעד:</u> נניח שהטענה נכונה לi ונוכיח ל$i+1$

$$\begin{aligned} j_{i+1} &\overset{\text{def}}{=} (j_i+(i+1)) \mod m \overset{\text{induction}}{\equiv} \left(j+\frac{1}{2}i+\frac{1}{2}i^2+(i+1)\right) \mod m \\ &= \left(j+\frac{1}{2}i+\frac{1}{2}(i+1)^2+\frac{1}{2}\right) \mod m \\ &= \left(j+\frac{1}{2}(i+1)+\frac{1}{2}(i+1)^2\right) \mod m \end{aligned}$$

כלומר הטענה נכונה עבור $i+1$, כנדרש

מ.ש.ל.א.☺

(ב) **צ"ל:** אין התנגשויות

הוכחה:

נניח בשלילה שהייתה התנגשות עבור $0\le k<l<m$ אזי

$$\begin{aligned} j_0+\frac{1}{2}\cdot\left(k+k^2\right) \mod m = j_k \equiv j_l &= j_0+\frac{1}{2}\cdot\left(l+l^2\right) \mod m \\ &\Rightarrow l^2-k^2+l-k\equiv 0 \mod 2m \\ &\Rightarrow (l-k)\cdot(l+k+1)\equiv 0 \mod 2m \end{aligned}$$

לכן $2^{i+1}=2\cdot 2^i=2m\mid (l-k)\cdot(l+k+1)$,

לכן נסמן $l-k=2^a\cdot b$, כאשר $i\ge a\ge 0$, $l+k+1=2^c\cdot d$, כאשר $i\ge c\ge 0$ (כדי להיות בטווח הנתון) ומתקיים $a+c>i+1$,

נשים לב כי $l+k+1\equiv l-k+1 \mod 2$,

לכן $l-k$ זוגי ו$l+k+1$ אי־זוגי או להפך, לכן בהכרח $a=0$ או $c=0$,

לכן

$$a+c=\max\{a,c\}\le i<i+1$$

בסתירה לכך ש$a+c>i+1$ בשביל ש$2^{i+1}=2\cdot 2^i=2m\mid (l-k)\cdot(l+k+1)$,

סתירה ולכן אין התנגשויות, לכן הוא ירוץ על כל תא לכל היותר פעם אחת

מ.ש.ל.ב.☺

2. **פתרון:**

(א) **צ"ל:** מבנה נתונים שמקיים $O(1)$ בהסתברות גדולה מ$\frac{1}{2}$

הוכחה:

נפתור עבור c כללי ונציב $c=1$, (סעיף ג')

נייישם מבנה נתונים שהוא perfect hashing שגודל המערך הראשון הוא $m=2^c\cdot N$,

עתה ממה שראינו בהרצאה מתקיים שהזיכרון הוא $O(2^c\cdot n)=O(n)$, כנדרש

וגם נשים לב כי

$$\begin{aligned}\mathbb{P}(\text{colision between 2}) &= \sum_{i=1}^{m}\mathbb{P}(\text{colision}|\text{ colision in i cell})\cdot\mathbb{P}(\text{colision in i cell})\\ &= \frac{1}{m}\cdot\sum_{i=1}^{m}\mathbb{P}[\text{colision}|\text{ colision in i cell}]\\ &= \frac{1}{m}\cdot\sum_{i=1}^{m}\frac{1}{n_i^2}\overset{\sum_{i=1}^{n}n_i=N}{\leq}\frac{1}{m\cdot N}\end{aligned}$$

לכן

$$\mathbb{P}(\text{colision})=\binom{n}{2}\cdot\mathbb{P}[\text{colision between 2}]=\binom{n}{2}\cdot\frac{1}{m\cdot N}\overset{m=N\cdot 2^c}{\leq}\frac{1}{2^{c+1}}<\frac{1}{2^c}$$

לכן

$$1-\mathbb{P}(\text{colision})\geq 1-\frac{1}{2^c}$$

נציב $c=1$, ונקבל את הסעיף המבוקש

מ.ש.ל.א.☺

(ב) **צ"ל:** מבנה נתונים שמקיים $O(1)$ בהסתברות גדולה מ$\frac{3}{4}$

הוכחה:

נציב במה שהוכחנו בסעיף א' $c=2$ ונקבל את המבוקש

מ.ש.ל.ב.☺

(ג) **צ"ל:** מבנה נתונים שמקיים $O(1)$ בהסתברות גדולה מ$1-\frac{1}{2^c}$

הוכחה:

הוכח בסעיף א' כטענת עזר

מ.ש.ל.ג.☺

3. **צ"ל:** אלגוריתם, פסאודו, והסבר

הוכחה:

האלגוריתם הוא לרוץ על DFS כך ש"נצבע" כל קודקוד בצבע אחד מתוך 2, כש2 סמוכים נרצה שיהיו בצבעים שונים, אם הצלחנו לצבוע ככה שאין 2 קרובים באותו צבע, אז ניתן ליצור ממשלה, אחרת בלתי אפשרי,
הצבעים הם כן ולא במקרה שלנו, הנה הפסאודו:

```
DFS(V, E):
    V[0].visited = True
    V[0].color = True
    for v in V:
```

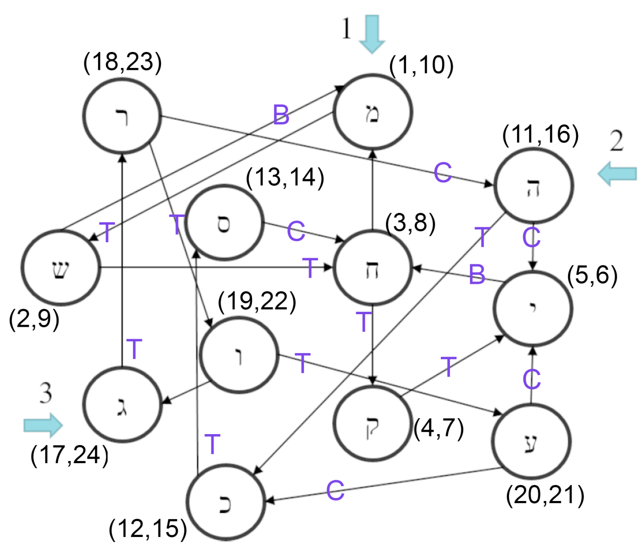
```
          if v.visited is False:
            if Explore(v, E) is False:
              return False
      return True
  Explore(v, E):
      for u in V such that (v,u) in E:
          if u.visited is False:
            u.visited = True
            u.color = !v.color
            if Explore(u, E) is False:
              return False
          #this is already explored, check if not same color
          else if u.color == v.color:
            return False
      return True
```

אם הצלחנו לצבוע ב2 צבעים, אז כל אלה שבאות הצבע יהיו ביחד, אם לא הצלחנו, אז אי אפשר לחלק ל2 גושים, כלומר אי אפשר ליצור ממשלה, כנדרש.

מ.ש.ל.☺

4. **פתרון:**

(א) **צ"ל:** ציירו את העץ
הוכחה:



מ.ש.ל.א.☺

(ב) **צ"ל:** הוכיחו יעילות אלגוריתם
הוכחה:
נשים לב כי בDFS נעשה $explore$ לכל קודקוד לכל היותר (ולפחות) פעם אחת,
ולכן זמן הריצה עבור כל קודקוד שווה ל

$$\Theta\left(1+\deg\left(v\right)\right)$$

לכן זמן הריצה הכולל שווה ל

$$\Theta\left(\sum_{v\in V}\left[1+\deg\left(v\right)\right]\right)=\Theta\left(\left|V\right|+2\left|E\right|\right)=\Theta\left(\left|V\right|+\left|E\right|\right)$$

מ.ש.ל.ב.☺

(ג) **צ"ל:** פסאודו עם $stack$
הוכחה:
זמן הריצה זהה לסעיף הקודם מאותו הנימוק של הקודקודים ושל סכומים, הנה הפסאודו:

```
DFS(V, E):
   stack = []
   root = V[0]
   stack.push(root)
   while stack is not empty:
     currentEdge = stack.pop()
     if currentEdge.visited is False:
       currentEdge.visited = True
         if u.visited is False:
           stack.push(u)
```

מ.ש.ל.ג.☺

(ד) **צ"ל:** הבדל עם $queue$
הוכחה:
אין הבדלים בין $queue$ ל$stack$ בדוגמא הנתונה כי בכל רגע נתון יש רק צלע אחת שניתן לעבור עליה.
אך ההבדל בעיקרון הוא שעם $queue$ אנחנו נעבור על כל השכנים במקביל במקום אחד אחרי השני,
אותו זמן ריצה מאותו הנימוק של סעיף ב'

מ.ש.ל.ד.☺

5. **פתרון:**

(א) **צ"ל:** האם יכול להתקבל מריצת DFS?
הוכחה:
לא! כי b אינו שכן של c

מ.ש.ל.א.☺

(ב) **צ"ל:** האם יכול להתקבל מריצת DFS?
הוכחה:
לא! כי ל a יש שכנים שלא בוקרו לפני שה$post$ הוגדר (במקרה זה תואר שלa אין שכנים כלל)

מ.ש.ל.ב.☺

(ג) **צ"ל:** האם יכול להתקבל מריצת DFS?
הוכחה:
כן! זאת בדיוק ההרצה שמתקבלת כשמתחילים מb, ומגרילים את האיבר הבא לאחר שהגיע לחוסר הצלחה ב$Explore$

מ.ש.ל.ג.☺

(ד) **צ"ל:** האם יכול להתקבל מריצת DFS?

הוכחה:

כן! זאת בדיוק ההרצה שמתקבלת כשמתחילים מc, ומגרילים את האיבר הבא לאחר שהגיע לחוסר הצלחה ב$Explore$

מ.ש.ל.ד.☺

(ה) **צ"ל:** האם יכול להתקבל מריצת DFS?

הוכחה:

לא! כי c אינו שכן של d

מ.ש.ל.ה.☺