פתרון תרגיל מספר 12 - דאסט

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

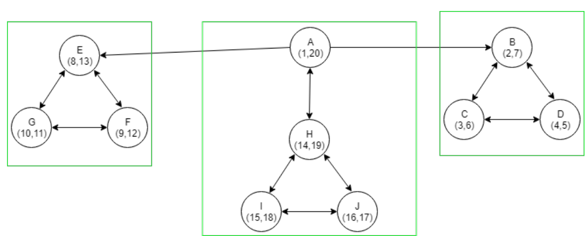
16 ביוני 2019

1. **צ"ל:** גרף + כיור

הוכחה:

נשים לב כי אם נעבור על זמני היציאה המקסימליים של כל אחד מרכיבי הקשירות בסדר יורד, למעשה נעבור עליהם בסדר הבא: נתחיל מרכיבי הקשירות אליהם אין צלעות.

עתה נשים לב כי גרף רכיבי הקשירות של G הוא ה-$Transpose$ של גרף רכיבי הקשירות T של G, ולכן עם נעבור על רכיבי הקשירות (אך הפעם על G) בסדר שמצאנו קודם, נקבל שתחילה נבקר ברכיבי T קשירות מהם אין צלעות יוצאות לרכיבים אחרים (כיורים). כלומר נימקנו למה נתחיל מכיורים, עתה דוגמא:



נשים לב שאם נתחיל מA, אז בריצה השנייה אם לא נרוץ בסדר יורד אז נקבל שB, A באותו רכיב קשירות על אף שהם לא.

מ.ש.ל.☺

2. **פתרון:**

(א) **צ"ל:** $O(n)$ במשעורין

הוכחה:

נסמן פעולה של $Stack$ בc, כאשר c הוא חסם על הפעולות של $stack$,

לכן עבור n פעולות יתקיים

$$O\left(\sum_{i=1}^{n} c+\sum_{i=1}^{\left\lfloor \frac{n}{k}\right\rfloor} k\cdot c\right)\le O\left(c\cdot n+c\cdot n\right)=O\left(2c\cdot n\right)=O\left(n\right)$$

מ.ש.ל.א.☺

(ב) **צ"ל:** $O(1)$ במשעורין

הוכחה:

ניתן שמות למחסניות s_1, s_2,

הוספה תהיה פשוט הוספה של איבר לs_1, שזה $O(1)$,

בדיקה שהטור ריק זה פשוט בדיקה ש2 המחסניות לא ריקות שזה $O(1)$,

הוצאה זה הוצאה מs_2 אם היא לא ריקה ($O(1)$), אחרת העברה של כל s_1 לs_2 ($O(k)$),

נשים לב שלאחר פעם אחת שהועבר מs_1 לs_2 שעלה $O(k)$ פעולות, נוכל לבצע k פעמים הוצאה ביעילות $O(1)$,

ולכן עבור k קטן זה יתנהג כמו $O(1)$ ועבור k גדול, נוכל לבצע הוצאה הרבה פעמים באופן יעיל, שהופך את הפונקציה ל$O(1)$ במשעורין

מ.ש.ל.ב.☺

3. **פתרון:**

(א) **צ"ל:** הדרגה r יש לפחות 2^r קודקודים
הוכחה:
נוכיח באינדוקציה על הדרגה
<u>בסיס:</u> בגובה 0 יש לנו קודקוד אחד ולכן יש לפחות $2^0 = 1$ קודקודים
<u>צעד:</u> נניח שהטענה נכונה ל$r - 1$ ונוכיח לr,
שיש לנו 2 עצים שדרגתם המקסימלית היא $k, l < r$, נאחדם,
אם העץ שנוצר הוא מדרגה קטנה מr, לא מעניין אותנו, אחרת הדרגה של העץ הנוצר הוא r,
כלומר $l = k = r - 1$, מהנחת האינדוקציה על l, k מתקיים כי יש לפחות 2^{r-1} קודקודים בכל אחד מהעצים,
אז בעץ שנוצר מאיחודם יש לפחות $2^{r-1} + 2^{r-1} = 2^r$

מ.ש.ל.א.☺

(ב) **צ"ל:** בדרגה r יש לכל היותר $\frac{n}{2^r}$ קודקודים
הוכחה:
מסעיף א' אנחנו יודעים שיש לכל תת עץ בגובה r לפחות 2^r קודקודים,
אז מספר המקסימלי של קודקודים בדרגה r יתקבל כאשר יש בדיוק 2^r בתת העץ,
כלומר יהיו בדיוק $\frac{n}{2^r}$ קודקודים בדרגה הr, לכן יש לכל היותר $\frac{n}{2^r}$ קודקודים בדרגה r

מ.ש.ל.ב.☺

(ג) **צ"ל:** $x.rank < x.parent.rank$
הוכחה:
מהתרגול אנחנו יודעים שתכונה זאת נשמרת בUnion By rank,
נשאר רק להראות שכאשר עושים $findset$ הpath comression משמר תכונה זאת,
נשים לב כי כל מה שהpath comression הוא קובע את הבן להצביע על האבא של האבא, לכן מהיות התכונה הזאת נשמרה עד כה מתקיים

$$x.rank < x.parent.rank < x.parent.parent.rank$$

ולכן התכונה נשמרת כאשר נגדיר את האב החדש של x להיות $parent.parent$

מ.ש.ל.ג.☺

(ד) **צ"ל:** בדרגה r יש לכל היותר $\frac{n}{2^r}$ קודקודים
הוכחה:
נוכיח באינדוקציה על הדרגה,
<u>בסיס:</u> $r = 0$, יש לכל היותר n קודקודים בדרגה 0 כי יש n קודקודים בכללי,
<u>צעד:</u> נניח שטענה נכונה לr ונוכיח ל$r + 1$,
נשים לב שכדאי ליצור קודקוד בדרגה $r + 1$ נדרשים 2 קודקודים בדרגה r,
לכן בדרגה $r + 1$ יש לכל היותר

$$\frac{\max\left\{\text{rank}\left(r\right)\right\}}{2} = \frac{\frac{n}{2^r}}{2} = \frac{n}{2^{r+1}}$$

כנדרש

מ.ש.ל.ד.☺

4. **פתרון:**

(א) **צ"ל:** $largest$
הוכחה:
נוסיף תכונה max לשורשי הקודקודים, וב$link$ נקבע את max להיות המקסימלי בין השורשים.
עתה $largest\left(i\right) = \text{find\_set}\left(i\right).root.max$

מ.ש.ל.א.☺

(ב) **צ"ל:** אלגוריתם אתר מלא
הוכחה:
נבצע את הפעולות הבאות על השורה הראשונה

i. ננסה ללכת למטה אם אפשר, אם כן ויש שם סט, נאחד עם הסט הנוכחי, אחרת נצרף לסט את הקודקוד למטה
ii. ננסה ללכת שמאלה אם אפשר, אם כן ויש שם סט, נאחד עם הסט הנוכחי, אחרת נצרף לסט את הקודקוד השמאלי
iii. ננסה ללכת ימינה אם אפשר, אם כן ויש שם סט, נאחד עם הסט הנוכחי, אחרת נצרף לסט את הקודקוד ימינה

אם בשלב כלשהו הגענו לשורה האחרונה, סיימנו והאתר הוא לא מסנן, אחרת אם רצנו על כל האלה בשורה הראשונה ולא הצלחנו, אז האתר הוא מסנן.
היעילות היא $O\left(n^2\right)$ כי במקרה הרוע רצים על כל האיברים ולכל איבר מגיעים לכל היותר 5 פעמים מכל שכניו.

מ.ש.ל.ב.☺

(ג) **צ"ל:** אלגוריתם משוואות
הוכחה:
נפתור בעיה זאת כמו הערים, תחילה נחבר בין כל 2 שעבורים להיות שווים,
לאחר מכן נעבור על כל 2 שלא אמורים להיות שווים ונבדוק שהם לא באותו רכיב קשירות.
אם הם באותו רכיב קשירות, אז זה לא פתיר וסיימנו.
אחרת אם עברנו על כל משוואות הלא שוות והכל היה בסדר, אז סיימנו גם כן כי אין שום תנאי שלא מתקיים ולכן זה פתיר.
יעילות:
נניח שיש k משוואות שווה,
החלק הראשון של החיבורים לוקח $O\left(k\right)$
החלק השני לוקח $O\left(k\right)$ לכל משוואה לא טובה ויש $m-k$ כאלה, כלומר

$$O\left(k\right)+O\left(k\right)\cdot\left(m-k\right)=O\left(k\cdot\left(m-k+1\right)\right)$$

במקרה הגרוע (פרבולה חיובית) $k=m$ ומתקיים $O\left(m\right)$

מ.ש.ל.ג.☺