פתרון תרגיל מספר 3 ־ פתרון בעיות באלגוריתמים

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

31 בדצמבר 2020

1. **הערה:** את הרעיון הכללי של הבונוס פתרתי ביחד עם עידן אורזך (הרעיון של הרצת מינימקס).
צ"ל: אלגוריתם לבעיה ב־ $O\left(n^2+m\right)$
הוכחה:
<u>עיבוד מוקדם:</u>

(א) נקבל את שורש העץ v ($O\left(1\right)$)

(ב) נוסיף קודקוד r שלא בגרף כך ש־ v יהיה הבן של r ($O\left(1\right)$)

(ג) נשמור את r בתור השורש של העץ ($O\left(1\right)$)

(ד) נריץ $LCA5$ על העץ ששורשו הוא r ונשמור את המבנה נתונים ($O\left(n\right)$)

כלומר זמן הריצה הוא $O\left(1\right)+O\left(1\right)+O\left(1\right)+O\left(n\right)=O\left(n\right)$ לעיבוד המוקדם

<u>שאילתה:</u>

(א) בהינתן u,v, נחשב את $LCA\left(u,v\right)=w$ ($O\left(1\right)$ בעזרת מבנה הנתונים השמור)

(ב) אם $w\neq r$ נחזיר שקיים מסלול בין u ל־ v ($O\left(1\right)$)

(ג) אחרת נחזיר שלא קיים מסלול ($O\left(1\right)$)

כלומר זמן הריצה הוא $O\left(1\right)+O\left(1\right)+O\left(1\right)=O\left(1\right)$ לכל שאילתה

<u>הסרת צלע:</u>

(א) בהינתן u,v, נמצא את הגובה של u ושל v (בלי הגבלת הכלליות המרחק של u מ־ r גדול מהמרחק של v מ־ r) ($O\left(n\right)$)

(ב) ננתק את v מלהצביע על u ($O\left(1\right)$)

(ג) נחבר את u להיות בן של r ($O\left(1\right)$)

(ד) נחשב מ־ 0 את $LCA5$ על העץ החדש ונשמור את המבנה במקום המבנה הקודם ($O\left(n\right)$)

כלומר זמן הריצה הוא $O\left(n\right)+O\left(1\right)+O\left(1\right)+O\left(n\right)=O\left(n\right)$ לכל הסרת צלע
כלומר ל־ n הסרות צלע ו־ m שאילתות נקבל זמן ריצה $O\left(n\right)\cdot n+O\left(1\right)\cdot m=O\left(n^2+m\right)$ כנדרש.

עתה מדוע האלגוריתם עובד?
הערה: נשים לב שהבנייה שלנו משמרת עץ בהינתן שבמקור היה עץ, (הוא נשאר קשיר כי כל מסלול בין u ל־ v ניתן להעביר בין u ל־ r ומ־ r ל־ v ועם $n-1$ צלעות ולכן הוא עץ)
נרצה להוכיח שה־ $LCA\left(u,v\right)=r$ בעץ שיצרנו אם"ם אין מסלול בין u ל־ v עץ המקורי.
$\Leftarrow$: נניח ש־ $LCA\left(u,v\right)=r$, נניח בשלילה שיש מסלול בין u ל־ v שנסמנו ב־ $u,v_1,\ldots,v_k,v$.
עתה מהיות שכל הצלעות הללו קיימות גם בעץ שיצרנו וגם שכל מסלול עובר ב־ $LCA\left(u,v\right)$, נסיק ש־ $\exists i\in\left[k\right]$ כך ש־ $v_i=r$.
אך בחרנו את r להיות קודקוד שלא קיים בעץ המקורי בסתירה לכך ש־ $u,v_1,\ldots,v_k,v$ מסלול בעץ המקורי.
$\Rightarrow$: נניח שאין מסלול בין u ל־ v עץ המקורי, נניח בשלילה ש־ $LCA\left(u,v\right)\neq r$,
כלומר קיים מסלול בין u ל־ v שלא עובר בשורש של העץ החדש שנסמנו ב־ $u,v_1,\ldots,v_k,v$.

נשים לב שהוספנו לעץ החדש רק צלעות שמכילות את r ומהיות $v_i \neq r$ לכל $i \in [k]$ נקבל שכל הצלעות האלה קיימות גם בעץ המקורי.
ולכן המסלול $u, v_1, \dots, v_k, v$ הוא גם מסלול בעץ המקורי בסתירה לכך שאין מסלול בין u ל־ v עץ המקורי.

הערה: את הרעיון הכללי של הבונוס פתרתי ביחד עם עידן אורזך (הרעיון של הרצת מינימקס).
בונוס ראשון:
נזכר באלגוריתם $minimax$ שראינו בקורסים אחרים, הרעיון הוא ליצור עץ של כל האפשרויות (בכל קודקוד יש לכל היותר 2 בנים והעומק הוא לכל היותר n ועלה הוא קודקוד שבו אין מהלך שהשחקן הנוכחי יכול לבצע) ולקחת מינימום \ מקסימום של הקודקודים כשהקודקוד הראשי הוא מקסימום.
ניתן לכל קודקוד ערך עם פונקצית ערך f שערכה תלוי בגובה של העץ שבו המשחק נגמר.
נבחר $f(k) = \begin{cases} n-k & \text{the root won} \\ 0 & \text{tie} \\ k-n & \text{the root lost} \end{cases}$, יש חשיבות לניצחון ואם יש ניצחון אז עם גובה מינימלי (כלומר פונקציה מונוטונית יורדת בגובה לשחקן שמנסה למקסם את רווחו).
הערה: הגובה נמדד מהשורש
ראינו בקורסים אחרים (לדוגמא AI), שבהינתן ו־ 2 השחקנים משחקים באופן אופטימלי, שניהם ייבחרו את המהלך האופטימלי לפי המינימיקס (כי המטרה היא להבטיח ניצחון ועם גובה של עץ כמה שיותר קטן).
נשים לב שבעץ המינימקס יש לכל היותר 2^n קודקודים והחישוב מינימום \ מקסימום בכל קודקוד לוקח $O(1)$ ולכן זמן הריצה של מציאת מהלך אופטימלי במינימקס לוקח $O(2^n)$.
נבצע את האלגוריתם הבא:

(א) כל עוד המשחק לא נגמר:
  i. נמצא מהלך אופטימלי מהמצב הנוכחי עם מינימקס
  ii. נשחק את המהלך הנוכחי
(ב) נחזיר את אורך המשחק

נשים לב שזמן הריצה הוא $O(2^n \cdot n) = O\left(2^{n+\log(n)}\right) = 2^{O(n)}$ (כי יש לכל היותר n מהלכים וכל איטרציה היא לוקחת זמן המינימקס שזה $O(2^n)$)
ומנכונות המינימקס כל שחקן משחק באופן אופטימלי ביחס לחוקי המשחק ולכן האלגוריתם יריץ בדיוק את המשחק האופטימלי.
בהינתן ויש כמה אפשרויות אופטימליות, לא משנה איזה מהמהלכים האופטימליים נעשה, המשחק ייגמר עם אותו ערך של הפונקציה ולכן ניתן לבחור אחד מהם באופן שרירותי.

מ.ש.ל.☺

2. **צ"ל:** אלגוריתם לבעיה ב־ $O(|V| + |E|)$
הוכחה:
הגדרנו בהרצאה $L(v) = \text{the minimal distance of } v \text{ to the root of the tree}$
הגדרנו בהרצאה $K(v) = \min\{L(w') \mid \text{w is a descendant of } v \text{ in the tree and } (w, w\prime) \text{ is a back edge}\}$
<u>אלגוריתם לבעיה:</u>

(א) נריץ את האלגוריתם שראינו בהרצאה לחישוב K, L לכל קודקוד ($O(|V| + |E|)$)
(ב) נסמן ב־ $G' = (V, E')$ את היער שנוצר מצלעות קדימה ב־ DFS ($O(1)$)
(ג) נחשב לכל $v \in V$: $Q[v] = \max_{(v,v') \in E'} K[v]$ ($O(|V| + |E|)$, בעזרת DFS שמחשבים מינימום על הבנים שלוקח $O(1)$ לכל בן)
(ד) נסמן את בשורש ב־ u, נחזיר $|\{v \mid Q[v] \geq L[v] \wedge v \neq u\} \cup \{u \mid u \text{ has 2 children or more}\}|$ ($O(|V|)$)

נשים לב שזמן הריצה הוא $O(|V| + |E|) + O(1) + O(|V| + |E|) + O(|V|) = O(|V| + |E|)$, כנדרש.
הערה: הרעיון הוא לשאול האם אפשר לקפוץ מכל תת עץ של v לקודקודים מעל v, המקום היחיד שבו זה בעייתי זה השושק שאין מעליו ולכן שואלים האם יש לו 2 בנים שונים.
עתה מדוע האלגוריתם עובד?
נסמן מסלול מ־ u ל־ v ב־ $u \to v$

עבור v שאינו השורש של העץ, נרצה להוכיח ש־ $Q(v) \geq L(v)$ אם"ם v מרכזי.
$\Leftarrow$: נניח ש־ $Q(v) \geq L(v)$ כלומר קיים בן v' כך ש־ $K(v') > L(v)$,
נניח בשלילה ש־ v לא מרכזי, כלומר הסרתו לא משנה את מספר רכיבי הקשירות של G.
נסמן את האבא של v ב־ v''. לכן לאחר הסרת v, בגרף שמתקבל קיים מ־ v' ל־ v'' מההנחה ש־ v לא מרכזי,
מהיות $K(v') \geq L(v)$, כל מסלול מ־ v' לא ירד מתחת לגובה $L(v)$, אך נשים לב כי יש מסלול מ־ v' ל־ v'' וגם $L(v'') < L(v')$, כלומר יש מסלול מ־ v' שמגיע לגובה מתחת ל־ $L(v)$ בסתירה לכך ש־ $K(v') \geq L(v)$
$\Rightarrow$: נניח ש־ v מרכזי, נניח בשלילה ש־ $Q(v) \leq L(v)$,
כלומר בהינתן u_1, u_2, מהיות הגרף המקורי קשיר, קיים מסלול $u_1 \to u_2$, נניח ש־v מופיע במסלול ונראה כיצד נחליפו. בהינתן $u_1, \ldots, w_1, v, w_2, \ldots, u_2$,
נבחין ש־ w_2 הוא בן של v ולכן מהיות $Q(v) \leq L(v)$ קיים w צאצא של w_2 וצלע (w, w') כך ש־ $L(w') < L(v)$
נשים לב כי $L(w') \leq L(w_1) < v < L(w_2) \leq L(w)$,
מהיות העץ קשיר קיים מסלול מ־ w' ל־ w_1 (שלא משתמש בצלעות v כי הוא נמוך משניהם)
מהיות העץ קשיר קיים מסלול מ־ w ל־ w_2 (שלא משתמש בצלעות v כי הוא גבוה משניהם וקיים מסלול כי w בתת עץ של w_2, פה משתמשים בעובדה שלקחנו מקסימום על כל הבנים)
ולכן נוכל להחליף את w_1, v, w_2 ב־ $w_1 \to w', w, \to w_2$ ובכך לקבל מסלול שלא כולל את v (להפעיל אבחנה זאת מספר פעמים).
ובסוף קיבלנו כי לכל זוג קודקודים u_1, u_2 שאינם v ניתן למצוא מסלול ללא צלעות v , ולכן הוא לא מרכזי בסתירה להנחה.

עתה נוכיח עבור השורש, השורש הוא מרכזי אם"ם יש לו לפחות 2 בנים.
$\Leftarrow$: השורש הוא מרכזי, נניח בשלילה שיש לו בן אחד,
לכן לכל זוג קודקודים יש מסלול בתוך התת עץ ללא מעבר בשורש.
כלומר השורש לא מרכזי, סתירה.
$\Rightarrow$: לשורש יש לפחות 2 בנים, נניח בשלילה שהשורש לא מרכזי,
נסמן 2 בנים של השורש ב־ u, v , מהיות והשורש לא מרכזי קיים מסלול בין u ל־ v ללא מעבר בשורש, אך זה בסתירה ליצירת עץ ה־ DFS (לא היינו פותחים את הבן השני דרך השורש אם היה אפשר להגיע אליו דרך הבן הראשון),
סתירה

כלומר הראנו שהתנאי לקודקוד מרכזי הוא $\{u \mid u \text{ has 2 children or more}\} \cup \{v \mid Q[v] \geq L[v] \wedge v \neq u\}$ וזה בדיוק מה שהחזרנו.

מ.ש.ל.☺