אלגוריתמים (67504) - פתרון תרגיל 5

שם: מיכאל גרינבאום, **ת.ז:** 211747639 ; **שם:** גיא לוי, **ת.ז:** 211744636

4 בדצמבר 2019

1. פתרון: נסמן את הקלט ב- $S = s_1 \dots s_n$, נגדיר $S_i = s_1 \dots s_i$
נחשב את $S^R = s_n \dots s_1$ לוקח $O(n)$, נסמן $S^R = x_1 \dots x_n$, נגדיר $S_i^R = x_1 \dots x_i$
עתה נריץ את האלגוריתם שראינו בתרגול על מציאת התת מחרוזת הארוכה ביותר המשותפת ל- S ול- S^R,

(א) <u>אוסף תתי הבעיות:</u> כמו שראינו בתרגול הבעיות הינן מציאת תת המחרוזת המשותפת של S_i, S_j^R, $0 \le \forall i \le n$, $0 \le \forall j \le n$

(ב) <u>נוסחת הרקורסיה:</u> כמו שראינו בתרגול הנוסחה היא:

$$M(i,j) = \begin{cases} 0 & (i=0) \vee (j=0) \\ M(i-1,j-1)+1 & s_i = x_j \\ \max\{M(i-1,j), M(i,j-1)\} & s_i \neq x_j \end{cases}$$

(ג) <u>הגדרת הטבלה:</u> כמו שראינו בתרגול, נגדיר טבלה M בגודל $(n+1) \times (n+1)$, נמלא את הטבלה $M(i,j)$ כמו בתרגול שורה שורה משמאל לימין.
<u>אופן חילוץ הפתרון:</u> כמו שראינו בתרגול, נשמור חץ לאיבר הקודם שבחרנו שממנו חישבנו את הפתרון, הפתרון נמצא בתא $M(n,n)$, לכן נתחיל מ- $M(n,n)$ ואז נרוץ על החצים וניצור את הפתרון האופטימלי. זמן חילוץ הפתרון הוא $O(n)$ בגלל שלכל היותר יש n איברים בפתרון.

(ד) <u>זמן ריצה:</u> כמו שראינו בתרגול, גודל הטבלה הוא $O(n^2)$ תאים ומילוי כל תא לוקח $O(1)$ וזמן חילוץ הפתרון הוא לכל היותר $O(n)$ ולכן זמן הריצה הוא

$$O(n^2) + O(n) = O(n^2)$$

2. נגדיר $X_i = x_1 \dots x_i$, $Y_j = y_1 \dots y_j$

(א) <u>אוסף תתי הבעיות:</u> מרחק העריכה המינימלי בין X_i ל- Y_j, $0 \le \forall i \le n$, $0 \le \forall j \le m$

(ב) <u>נוסחת הרקורסיה:</u> נגדיר $M(i,j)$ מרחק העריכה עבור תת הבעיה X_i, Y_j.

i. אם $i = 0$ אז צריך להוסיף ל- X_i את כל התווים של Y_j או להוריד מ- Y_j את כל האותיות ולכן $M(0,j) = j$.

ii. אם $j = 0$ אז צריך להוסיף ל- Y_j את כל התווים של X_i או להוריד מ- X_i את כל האותיות ולכן $M(i,0) = i$.

iii. אם $x_i = y_j$, אז לא צריך לשנות את x_i או y_j, נשאר לטפל בבעיה X_{i-1}, Y_{j-1}, ולכן נשאר לפתור את הבעיה $M(i-1,j-1)$.

iv. אם $x_i \neq y_j$, יש 3 אפשרויות למה לעשות:

א'. אם נוריד את x_i, נשאר לטפל בבעיה X_{i-1}, Y_j, כלומר $M(i-1,j)$.

ב'. אם נוסיף את y_j ל- X אחרי x_i, נצטרך לטפל ב- $Y_j, X_i \cup \{y_j\}$ ושניהם מסתיימים ב- y_j, לכן נוכל לטפל רק באותיות השונות שהם X_i, Y_{j-1}, כלומר נשאר לטפל ב- X_i, Y_{j-1} ולכן הנוסחא היא $M(i,j-1)$.

ג'. אם נחליף את y_j ב- x_i, נצטרך לטפל ב- $X_i, Y_{j-1} \cup \{x_i\}$ ושניהם מסתיימים ב- x_i, לכן נוכל לטפל רק באותיות השונות שהם X_{i-1}, Y_{j-1}, כלומר נשאר לטפל ב- X_i, Y_{j-1} ולכן הנוסחא היא $M(i-1,j-1)$.

לכן היות ואנחנו רוצים את הפתרון המינימלי בין ה- 3 עם הפעולה שבחרנו, הנוסחא למקרה הזה היא

$$1 + \min\{M(i-1,j), M(i,j-1), M(i-1,j-1)\}$$

לכן נוסחת הרקורסיה היא

$$M(i,j)=\begin{cases} i+j & (i=0)\vee(j=0)\\ M(i-1,j-1) & x_i=y_j\\ 1+\min\{M(i-1,j),M(i,j-1),M(i-1,j-1)\} & x_i\neq y_j\end{cases}$$

(ג) הגדרת הטבלה: נגדיר טבלה M בגודל $(n+1)\times(m+1)$, נמלא את הטבלה $M(i,j)$ מ־ $j=0$ עד ל־ $j=m$, כאשר לכל j, נמלא מ־ $i=0$ עד ל־ $i=n$.
אופן חילוץ הפתרון: הפתרון נמצא ב־ $M(n,m)$ ולכן זמן חילוץ הפתרון הוא $O(1)$.

(ד) זמן ריצה: גודל הטבלה הוא $O(nm)$ תאים ומילוי כל תא לוקח בדיקת 3 תאים קודמים ולכן $O(1)$ וזמן חילוץ הפתרון הוא $O(1)$ ולכן זמן הריצה הוא

$$O(nm)+O(1)=O(nm)$$

3. פתרון:

(א) תתי בעיות: לכל קודקוד $i\in V$ נגדיר את $f(i)$ להיות הקבוצה הבלתי תלויה בעלת המשקל המקסימלי בתת העץ ששורשו i (ו־ i יכול להיות חלק ממנה).

(ב) נוסחת רקורסיה:

i. מקרה בסיס: לכל עלה i המשקל המקסימלי הוא משקל העלה $f(i)=w(i)$.

ii. מקרה כללי: כל שלב צריך להחליט אם לקחת את האב או לא – אם לקחנו לא נוכל לקחת את הקבוצות הבלתי תלויות עם משקל מקסימלי בתתי העץ ששורשיהם הם מילדיו ולכן ניקח את האב יחד עם הקבוצות של נכדיו. אחרת, בלי האבא ניקח את פתרון ילדיו.

נוסחא:

$$f(i)=\max\left\{w(i)+\sum_{j\text{ is grandchild of }i}w(j),\sum_{j\text{ is child of }i}w(j)\right\}$$

(ג) בניית טבלה: במקום שימוש בטבלה, נשתמש בעץ עם $|V|$ קודקודים ואותו מבנה (צלעות בין קודקודים) של T הנתון. נמלא את העץ במעבר רקורסיבי על הילדים ובסוף על השורש (DFS Postorder) בערכים $f(i)$ שחישבנו בסעיף הקודם, ערך בוליאני אם לקחנו את קודקוד זה או לא וחץ לקודקוד הבא שבחרנו (בן או נכד בהתאמה) – נשים לב כי ככה לפני שנמלא את צומת i בעץ שאנו ממלאים, נמלא את כל הקודקודים בתת העץ שזהו שורשו (או שהוא עלה ואז אין לו ילדים בהם הוא תלוי).
אופן החילוץ: נגדיר קבוצה ריקה A. בסוף גם הבנים של העץ שמילאנו מלאים, נבחר את הבן עם הקבוצה בלתי תלויה עם משקל מקסימלי ונבנה קבוצה של קודקודים באופן הבא: אם הקודקוד נבחר (התנאי הבוליאני השמור בתוכו אומר לקחת), נכניס אותו לקבוצה A ונקפוץ לנכד השמור, אחרת נקפוץ לבן השמור. נחזור על התהליך עד אשר נסיים בעלה ובסוף נחזיר את A כפתרון.

(ד) זמן ריצה: אנו מבצעים מעבר DFS על העץ וכן החישוב על כל קודקוד לוקח זמן בחישוב של שלושה קודקודים (עצמו, אביו וסבו) – ככה שכל פעולה על כל קודקוד עדיין עולה $O(1)$ וה־ DFS עדיין עולה $O(n)$. חילוץ הפתרון עובר על הקודקודים מהשורש לעץ במקרה הכי גרוע וחסום מלמעלה ב־ $O(n)$ (כל קודקוד נעשה בדיקה ב־ $O(1)$ של התנאי הבוליאני השמור בו וקפיצה לקודקוד הבא) – סך הכל $\boxed{O(n)}$.

4. פתרון: ראשית נסמן $V=(v_0,v_1,\dots,v_n)$.

(א) תתי בעיות: לכל $v\in V$ נמצא את הקודקודים $v_{i(1)},\dots,v_{i(j)}$ שיוצרים מסלול ב־ G, $\left(v_{i(1)},\dots,v_{i(j)}\right)$ (כלומר הצלעות בין קודקודים סמוכים קיימות ב־ E) כך ש־ $v_{i_1}=v_0$ ו־ $v_{i(j)}=v$ וכן

$$\forall 1\le t\le j-1:\sigma\left(\left(v_{i(t)},v_{i(t+1)}\right)\right)=s_t$$

וגם שהמסלול בעל משקל מקסימלי.

(ב) נוסחת רקורסיה:

i. מקרה בסיס: לכל קודקוד בעל צלע e עם v_0 כך ש־ $\sigma(e)=s_1$ נגדיר ערך $w(e)$ (זה המשקל המקסימלי של מסלול ביניהם), לשאר הקודקודים נגדיר $-\infty$.

ii. מקרה כללי: כל שלב, עבור התו ה־ i במחרוזת לכל קודקוד $v_j \in V$ באופן הבא: אם קיים קודקוד היוצר איתו צלע שנכנסת אליו ב־ G שמחזירה עם σ את s_i, נמצא את הקודקוד שמחזיר עבור התו ה־ $i-1$ את הערך המקסימלי ונכניס לתא הנוכחי את ערך זה שמצאנו ועוד משקל הצלע ביניהם, אחרת $-\infty$:

הנוסחא:

$$f(i,j) = \begin{cases} \begin{cases} w((v_0,v_j)) & (v_0,v_j)\in E \wedge \sigma((v_0,v_j)) = s_1 \\ -\infty & \text{otherwise} \end{cases} & i=1 \\ \max\limits_{\substack{0\le t\le n \\ (v_t,v_j)\in E \\ \sigma((v_t,v_j)) = s_i}} \{f(i-1,t) + w((v_t,v_j))\} & \text{otherwise} \end{cases}$$

נשים לב כי אם הקבוצה בתוך ה־ max ריקה, נגדיר שיחזור הערך $-\infty$.

(ג) <u>בניית הטבלה:</u> נבנה טבלה בגודל $k\times(n+1)$.
<u>מילוי הטבלה:</u> נמלא את השורה הראשונה לפי נוסחת הרקורסיה, בצורה שרירותית נבחר משמאל לימין, ונמשיך לשורה הבאה ונמלא אותה גם לפי הנוסחא וכן הלאה עד שנגיע לשורה האחרונה. מלבד הערך של משקל המסלול נשמור גם מצביע לקודקוד הקודם במסלול.
<u>אופן החילוץ:</u> נגדיר A קבוצה ריקה. נמצא את המקסימום של השורה העליונה ־ אם שווה ל־ $-\infty$ נסיק שאין מסלול המקיים את הדרוש ונחזיר קבוצה ריקה. אחרת מצאנו ערך מקסימלי ממשי באחד התאים בשורה העליונה, נכניס את הקודקוד המתאים לתא (אם עמודה j אז קודקוד v_j) ונקפוץ לתא הבא אליו הוא מצביע שורה מתחת ־ כשנגיע לשורה הראשונה נחזיר את A כפתרון הדרוש.

(ד) <u>זמן ריצה:</u> נשים לב כי כל מילוי השורה הראשונה לוקח $O(n+1)$, ושאר $k-1$ השורות לוקח $O\left((n+1)^2\right)$ (לכל תא לעבור על כל התאים של השורה הקודמת כדי למצוא מקסימום) ־ סך הכל $O\left(k\cdot(n+1)^2\right)$. זמן חילוץ הפתרון: לוקח $O(n+1)$ למצוא את המקסימום של השורה העליונה, ומאז מעבר משורה לשורה ־ כלומר סך הכל $O(n+1+k)$ ולכן זמן הריצה שלנו סך הכל $\boxed{O\left(k\cdot|V|^2\right)}$.

5. פתרון:

(א) <u>אוסף תתי הבעיות:</u> מציאת חלוקה מינימלית של $w_1,\ldots,w_i$ ל־ j קבוצות $0\le\forall i\le n$, $0\le\forall j\le m$,

(ב) <u>נוסחת הרקורסיה:</u> נגדיר את נוסחת הרקורסיה עבור ההתאמה $M(i,j)$:

i. אם $i=0$, אז נגדיר $M(0,j)=0$ כי המחיר הוא 0 בכללי כי שום דבר לא נלקח.

ii. אם $j=1$, נהיה חייבים לקחת את כל הקבוצה ולכן $M(i,1)=\sum_{l=1}^{i} w_l$

iii. **הערה:** לא משתלם שתהיה קבוצה ריקה כי ככה המקסימום יהיה רק יותר גדול.
לכן, נחפש $k\le i$ אופטימלי שעבורו החלוקה של הקטע האחרון שאינו ריק עם ה־ $j-1$ קטעים אחרים היא אופטימלית, כלומר

$$\min_{1\le k\le i}\left\{\max\left\{M(k-1,j-1),\sum_{l=k}^{i} w_l\right\}\right\}$$

לכן נקבל שנוסחת הרקורסיה היא:

$$M(i,j) = \begin{cases} 0 & i=0 \\ \sum_{l=1}^{i} w_l & j=1 \\ \min_{1\le k\le i}\left\{\max\left\{M(k-1,j-1),\sum_{l=k}^{i} w_l\right\}\right\} & else \end{cases}$$

נוסיף לכל $M(i,j)$ את מיקום החלוקה האופטימלי שנמצא שנסמנו ב־ $g(i,j)=i-k$ (עבור ה־ k האופטימלי שנמצא), ואם לא נמצא נגדיר $g(i,j)=-1$.

(ג) <u>הגדרת הטבלה:</u> נגדיר טבלה M בגודל $(n+1)\times m$, נמלא את הטבלה $M(i,j)$ לפי מספר החלוקות j, נרוץ מ־ $j=1$ עד ל־ $j=m$, והמילוי בתוך החלוקה j נרוץ מ־ $i=0$ עד ל־ $i=n$.
<u>אופן חילוץ הפתרון:</u> האורך המינימלי נמצא ב־ $M(n,m)$, אם נרצה למצוא את החלוקה עצמה, נלך לפי $g(i,j)$ עד שנגיע ל־ $g(i,j)=-1$. נשים לב ש־ $j\le m$ ולכן זמן חילוץ הפתרון הוא $O(m)$ כי מבצעים $O(1)$ פעולות בכל תא וניגשים ל־ m תאים.

הערה: נשים לב שאם נרוץ נאיבית עם האלגוריתם שנכתב, זמן הריצה לכל תא יהיה $O\left(n^2\right)$ בגלל החישוב של $\sum_{l=k}^{i} w_l$ לכל תא, נוכל לייעל אותו אם בחישוב של התא, נרוץ מ־ $k=i$ ל־ $k=1$ ונשמור את הסכום עד כה וכל פעם נצטרך רק להוסיף את w_k לסכום. אם הייעול הזה זמן מילוי תא הוא $O\left(n\right)$ כי עוברים על לכל היותר על n תאים שהאינדקס k רץ עליהם.

(ד) זמן ריצה: גודל הטבלה הוא $O\left(nm\right)$ וזמן מילוי כל תא בטבלה הוא $O\left(n\right)$ וזמן חילוץ הפתרון הוא $O\left(m\right)$ ולכן זמן הריצה הוא

$$O\left(n^2 m\right) + O\left(m\right) = O\left(n^2 m\right)$$