פתרון בוחן algo122 – 18.5.12

שאלה 1:

בדומה למבנה ההוכחה עבור נכונות אלגוריתם $Dijkstra$ נוכיח את הטענה באינדוקציה על סדר הכנסת הקודקודים ל - S.

בסיס:הקודקוד הראשון המוכנס ל-S הוא תמיד s ובעת הכנסתו $d(s) = 0$ אזי הטענה מתקיימת באופן ריק.
הנחה: נניח כי הטענה נכונה לכל קודקוד u שהוכנס ל - S לפני v ונוכיח עבור v.

צעד: נביט בקודקוד v שהינו הקודקוד הבא שהאלגוריתם בוחר להכניס ל-S, רגע לפני הכנסתו. נוכיח כי קיים מסלול זול ביותר P מ-s ל-v כך שכל קודקודיו מלבד v שייכים בשלב זה ל-S.
נחלק לשני מקרים:
1.אם v לא נגיש מ-S אזי, עפ"י ההגדרה $\delta(s,v) = \infty$ ולכן אין מה להוכיח.
2. v נגיש מ-S: אזי עפ"י ההגדרה $\delta(s,v) < \infty$ ולכן מנכונות אלגוריתם $Dijkstra$, בעת בחירת v מתקיים $d(v) = \delta(s,v) < \infty$. מכאן שבהכרח כבר בוצע בשלב זה $Relax(u,v,w)$ עם קודקוד כלשהו u. יהא u הקודקוד האחרון, לפני בחירת v כך שעבורו היה עדכון במהלך ביצוע $Relax(u,v,w)$ ולכן מתקיים כעת: $d(v) = d(u) + w(u,v)$.
ידוע ממבנה האלגוריתם כי הדבר מעיד בהכרח שבשלב זה $u \in S$. מהאינווריאנטה של אלגוריתם $Dijkstra$ שהוכחה בכיתה ידוע כי בעת שהוכנס u ל-S התקיים : $d(u) = \delta(s,u)$ ולא היה שינוי בערכו של $d(u)$ מאז. כמו כן, מהנחת האינדוקציה הטענה מתקיימת עבור u, לכן בהכרח קיים מסלול זול ביותר P' מ-s ל-u כך שכל קודקודיו ב-S. בסה"כ קיבלנו $cost(P') = d(u)$. כיוון שהאלגוריתם בוחר ב-v להיות הקודקוד הבא שייכנס ל-S אזי בשלב זה מתקיים $d(v) = \delta(s,v)$ ובסה"כ $\delta(s,v) = \delta(s,u) + w(u,v)$. כעת אם נגדיר את P כשרשור v ל- P' נקבל כי:
$cost(P) = cost(P') + w(u,v) = \delta(s,u) + w(u,v) = \delta(s,v)$. כלומר P הינו מסלול זול ביותר מ-s ל-v שכל קודקודיו ב-S כנדרש.

שאלה 2:

סעיף א'

הטענה נכונה. **הוכחה**:
נניח בשלילה שקיים עץ פורש מינימום $T = (V, E_T)$ של G כך ש- $e \notin E_T$. נביט בגרף $H = (V, E_T \cup \{e\})$. ממשפט 2 בדפי העזר H מכיל מעגל יחיד. תהא $e' \neq e$ צלע כלשהי במעגל.
כיוון ש-e הינה הצלע היחידה ב-G בעלת משקל שלילי אזי בהכרח (*) $w(e) < w(e')$. ממשפט 2 נקבל בנוסף כי $T' = (V, \{E_T \setminus \{e'\}\} \cup \{e\}$ הינו עץ פורש מינימום של G. מ- (*):
$w(T') = w(T) - w(e') + w(e) < w(T)$ בסתירה להנחה על T.

סעיף ב'

נציע שני אלגוריתמים אפשריים:

אלגוריתם 1:

1. הרץ אלגוריתם למציאת עץ פורש מינימום על G ויהא $T = (V, E_T)$ עץ הפלט.

2. $\mathrm{E}' \leftarrow \mathrm{E_T}$
3. עבור כל $e \in E \backslash E_T$:
   3.1. אם $w(e) < 0$:
      3.1.1. $\mathrm{E}' \leftarrow \mathrm{E_T} \cup \{\mathrm{e}\}$.
4. החזר $\mathrm{G}' = (V, E')$.

ניתוח זמן ריצה:

- מציאת *MST* - נניח ע"י פרים: $O(|V|\log(|V|) + |E|)$
- הלולאה בשלב 3: $O(|E|)$.
- סה"כ: $O(|V|\log(|V|) + |E|)$.

אלגוריתם 2:

1. חשב $G' = (V, E')$. באשר: $E' = \{e \in E | w(e) < 0\}$.
2. חשב $G'' = (V'', E'')$. באשר
   - קודקודי G'' מייצגים את הרכיבים הקשירים היטב של G'.
   - צלעות G'' הינם צלעות מ-E בעלות משקל חיובי תחת w אשר מחברות ב-G קודקוד השייך לרכיב קשירות אחד של G' לרכיב קשירות אחר. אם יש יותר מאחת כזו בין אותם רכיבי קשירות הצלע הנבחרת תהיה זו בעלת המשקל המינימאלי מביניהן.
3. הגדר פונקצית משקל: $w'': E'' \rightarrow \mathbb{R}$ כך ש:
   $w''((X,Y)) = w\big((u,v)\big)\ s.t\ (u,v)\ made\ (X,Y)\ to\ appear\ in\ E''$.
4. מצא עץ פורש מינימום של G'' תחת w'', יהא $T = (V, E_T)$ העץ שנמצא.
5. החזר את הגרף: $G''' = (V, E_T \cup E')$.

ניתוח זמן ריצה:

- חישוב G' - סריקה יחידה של קשתות וקודקודי G - $O(|V| + |E|)$.
- מציאת רכיבים קשירים בגרף לא מכוון באמצעות DFS - $O(|V| + |E'|) = O(|V| + |E|)$.
- בניית E'' - סריקה יחידה של קשתות וקודקודי G - $O(|V| + |E|)$.
- מציאת MST ב- G''- נניח ע"י קרוסקל: $O(|V''|\log(|V''|) + |E''|) = O(|V|\log(|V|) + |E|)$.
- בניית G''' - $O(|V| + |E_T \cup E'|) = O(|V| + |E|))$
- סה"כ: $O(|V|\log(|V|) + |E|)$.

שאלה 3:

סעיף א'

יהא r שורש העץ. נניח בשלילה כי קיים פיתרון חוקי לבעיה $I \subseteq V$ כך שמתקיים: $r \notin I$ וגם קיים $v \in V$ כך ש: $(r,v) \in E$ אך $v \notin I$. אך נשים לב כי אם כך הדבר אזי שתי הקצוות של הצלע $(r,v) \in E$ אינן שייכות ל- I בסתירה להנחה כי הוא פיתרון חוקי.

סעיף ב'

תתי בעיות

לכל $v \in T$ נגדיר :

$\mathrm{OPT}(v)$ - משקלה של קבוצת קודקודים חוקית (פתרון חוקי) בעלת משקל מינימאלי עבור תת העץ המושרש ב - v.

לכל $v \in T$ יהא T_v - תת הבעיה עבור העץ הנטוע בקודקוד v.

מיקום הפיתרון לבעיה כולה יימצא ב - $\mathrm{OPT}(root(T))$.

<u>נוסחת מבנה</u>:

$$\mathrm{OPT}(v) = Min\left\{\sum_{u\in Sons(v)} w(u) + \sum_{u\in Grandsons(v)} OPT(u), w(v) + \sum_{u\in Sons(v)} OPT(u)\right\}$$

באשר $sons(v)$ הינה קבוצת הבנים הישירים של v. $Grandsons(v)$ הינה קבוצת הנכדים של v (בנים ישירים של בניו של v). נשים לב כי ייתכן ש - $sons(v)$ ו-$Grandsons(v)$ יהיו קבוצות ריקות ובמקרה זה ערך ה- OPT המתאים הוא 0.

<u>סעיף ג'</u>

מרחב הפיתרונות : Sol - כל קבוצות הקודקודים החוקיות המוכלות בתת העץ T_v המושרש ב - v.

נחלק את מרחב הפיתרונות באופן הבא:

- S_1 - קבוצת כל הפתרונות ב- Sol אשר מכילים את v.
- S_2 - קבוצת כל הפתרונות ב- Sol אשר לא מכילים את v.

נגדיר $O^*(S_i) = Min_{I\in S_i}\{cost(L)\}$, $i \in \{1,2\}$.

כיסוי: יהא I פיתרון חוקי, אם הוא מכיל את v הוא שייך ל - S_1. אחרת ל- S_2.

לכן מתקיים (מסקנה לצורת נוסחת המבנה):

$$OPT(v) = \mathrm{Min}\,\{O^*(S_1), O^*(S_2)\}$$

נותר להוכיח את הטענות הבאות:

- טענת עזר 1: $O^*(S_1) = w(v) + \sum_{u\in Sons(v)} OPT(u)$
- טענת עזר 2: $O^*(S_2) = \sum_{u\in Sons(v)} w(u) + \sum_{u\in Grandsons(v)} OPT(u)$

<u>הוכחת ט.ע 1</u>:

נניח $\{u_1, \dots, u_k\} = Sons(v)$. (ייתכן שקבוצה זו ריקה).

$\leq$

יהיה P_i $(1 \leq i \leq k)$ פיתרון חוקי עבור T_{u_i} כך ש: $cost\,(P_i) = OPT(u_i)$. נשים לב שכיוון שמדובר בעץ אזי כל תתי הבעיות $T_{u_1}, \dots, T_{u_k}$ זרות בזוגות. מכאן שנוכל לבנות פתרון P לעץ כולו כך ש: $P = \{\bigcup_{i=1}^{k} P_i\} \cup \{v\}$, נבחין כי כיוון שכל אחד מתתי הפתרונות של תתי העצים שנטועים בבניו של v חוקיים אזי P הינו פתרון חוקי לעץ שמכיל את v. מכאן ש - $P \in S_1$ ולכן מהגדרת $O^*(S_1)$ מתקיים הנדרש:

$$O^*(S_1) \leq cost(P) = w_j + \sum_{u_i\in Sons(V)} cost(P_i) = w_j + \sum_{u_i\in sons(V)} OPT(P_i)$$

$\geq$

יהא P פיתרון חוקי לבעיה המכיל את v כך ש- $cost(P) = O^*(S_1)$. P מכיל מלבד v קודקודים השייכים לתתי העצים $T_{u_1}, \dots, T_{u_k}$. נסמן :

$\forall 1 \leq i \leq k\colon\ P_i = \{x \in T_{u_i} \wedge x \in P\}$, נשים לב כי כי k תתי פתרונות אלו מהווים פתרונות חוקיים לתתי הבעיות המתאימות כתתי פיתרונות של הפיתרון החוקי P, תתי העצים האחרים לא רלוונטיים לבעיה ואילו הוצאת השורש המקורי מהפיתרון ודאי לא פוגע בחוקיות. מכאן שלכל i נובע מהגדרת $OPT(u_i)$ כי:

$cost(P_i) \geq OPT(u_i)$. כעת, מהעובדה כי k תתי הבעיות זרות בזוגות נוכל לקבוע:

$$O^*(S_1) = cost(P) = [\sum_{i=1}^{k} cost(P_i)] + w(v) \geq \sum_{u\in Sons(v)} OPT(u) + w(v) \ .$$

הוכחת ט.ע 2:

נניח $Sons(v) = \{u_1, \dots, u_k\}$. (ייתכן שקבוצה זו ריקה).

נניח $Grandsons(v) = \{t_1, \dots, t_m\}$. (ייתכן שקבוצה זו ריקה).

$\leq$

יהיה P_i $(1 \leq i \leq m)$ פיתרון עבור T_{t_i} כך ש: $cost(P_i) = OPT(t_i)$. נשים לב שכיוון שמדובר בעץ אזי כל תתי הבעיות $T_{t_1}, \dots, T_{t_k}$ זרות בזוגות. נבחין כי כל אחד מתתי הפתרונות של תתי העצים שנטועים בנכדיו של v חוקיים, כמו כן - $v \notin P$ בהכרח ולכן מסעיף א' $Sons(v) \subseteq P$. מכאן שנוכל לבנות פתרון חוקי P לעץ כולו כך ש:

$P = Sons(v) \cup \{\bigcup_{i=1}^{m} P_i\}$. אזי P הינו פתרון חוקי לעץ שאינו מכיל את v. מכאן ש - $P \in S_2$ ולכן מהגדרת $O^*(S_2)$ מתקיים הנדרש:

$$O^*(S_2) \leq cost(P) = \sum_{u \in Sons(v)} w(u) + \sum_{t_i \in Grandsons(V)} cost(t_i)$$

$$= \sum_{u \in Sons(v)} w(u) + \sum_{t_i \in Grandsons(V)} OPT(t_i)$$

$\geq$

יהא P פיתרון חוקי לבעיה שאינו מכיל את v כך ש- $cost(P) = O^*(S_2)$. P פיתרון חוקי שאינו מכיל v ולכן מסעיף א' הוא מכיל בהכרח את כל בניו של v ולכן מלבדם הוא מכיל אך ורק קודקודים השייכים לתתי העצים $T_{t_1}, \dots, T_{t_k}$. נסמן :

$\forall 1 \leq \mathrm{i} \leq m$: $P_i = \{\mathrm{x} \in T_{t_i} \wedge x \in P\}$, נשים לב כי m תתי פתרונות אלו מהווים פתרונות חוקיים לתתי הבעיות המתאימות. מכאן שלכל i נובע מהגדרת $OPT(t_i)$ כי:

$cost(P_i) \geq OPT(t_i)$. כעת, מהעובדה ש- m תתי הבעיות זרות בזוגות נוכל לקבוע:

$$O^*(S_2) = cost(P) = \sum_{u \in Sons(v)} w(u) + \sum_{t_i \in Grandsons(V)} cost(t_i)$$

$$\geq \sum_{u \in Sons(v)} w(u) + \sum_{t_i \in Grandsons(V)} OPT(t_i)$$

כנדרש.

סעיף ד'

אלגוריתם:

$\boldsymbol{Alg(T)}$

1. *Traverse T in post-order - Let v be the current vertex:*
   $M[v] = Min\{\sum_{u \in Sons(v)} w(u) + \sum_{u \in Grandsons(v)} M[u], w(v) + \sum_{u \in Sons(v)} M[v]\}$
2. *return* $M[root(T)]$

זמן ריצה:

סריקת העץ לוקחת $O(|V|)$. נשים לב שכל קודקוד הוא בן של קודקוד יחיד ונכד של קודקוד יחיד ולכן ייבדק לכל היותר פעמיים. בסה"כ - $O(|V|)$.

סעיף ה'

האלגוריתם:

אם $M[root(T)] = w(root(T)) + \sum_{u \in Sons(root(T))} M[root(T_u)]$ החזר "כן", אחרת החזר "לא".

זמן ריצה: העץ הינו עץ בינארי, לכן יש לשורש לכל היותר 2 בנים ומכאן ההשוואה לוקחת זמן קבוע.