מבני נתונים ואלגוריתמים – תרגיל 5

שאלה 1

יש רשימה של n משימות שצריכות להתבצע. משימה i דורשת t_i דקות כדי לבצע אותה. בנוסף יש m תלויות בין המשימות. כל התלויות הן מהצורה "משימה i חייבת להתבצע לאחר משימה j". תלות כזו תיוצג ע"י הזוג (i, j). ניתן להניח כי אין סתירה בין התלויות.

כתבו אלגוריתם המקבל את n, את הזמנים $\{t_i\}_{i=1}^{n}$ ואת התלויות $\{(i_k, j_k)\}_{k=1}^{m}$ ומחזיר את הזמן המינימלי הדרוש לביצוע כל המשימות (כאשר לרשותכם כוח אדם ללא הגבלה). זמן הריצה חייב להיות פולינומיאלי ב-n, m. הוכיחו את נכונות האלגוריתם.

לדוגמא: אם יש חמש משימות

1. לפרוש מפה על השולחן – 2 דקות
2. לשים צלחות על השולחן – 5 דקות
3. לשים כוסות וסכו"ם על השולחן – 6 דקות
4. לבשל אוכל – 20 דקות
5. לשים את האוכל על הצלחת – 3 דקות

והתלויות הן: 2 אחרי 1, 3 אחרי 1, 5 אחרי 4, 5 אחרי 2.

אז הזמן המינימלי הדרוש הוא 23 דקות – מתחילים לעשות את 4 (20 דקות) ובמקביל עושים את 1 (2 דקות) ואח"כ את 3 (6 דקות) ו-2 (5 דקות) בו זמנית. לבסוף עושים את 5 (3 דקות).

פיתרון

נבנה גרף $G = (V, E)$ בו $V = \{1, \dots, n\}$ (קבוצת המשימות) ו-$E = \{(j_k, i_k)\}_{k=1}^{m}$, כלומר, $(u, v) \in E$ אם"ם u חייב להתבצע לפני v. לכל קודקוד $v \in V$ נחזיק משתנה:

- $StartTime(v)$ - הזמן המינימלי בו אפשר להתחיל את המשימה. בהתחלה הוא יאותחל ל-0 עבור כל הקודקודים.

בנוסף, נחזיק משתנה בשם $FinishTime$.

האלגוריתם יהיה: (קלט: $\{t_i\}_{i=1}^{n}$ ו-$\{(i_k, j_k)\}_{k=1}^{m}$)

1. $FinishTime \leftarrow 0$
2. בנה את הגרף $G = (V, E)$
3. כל עוד קיים קודקוד $v \in V$:
   a. מצא קודקוד $v \in V$ כך שאין קשתות שנכנסות ל-V.[1]
   b. לכל $u \in V$ כך ש-$(v, u) \in E$:
      i. $StartTime(u) \leftarrow \max\{StartTime(u), StartTime(v) + t_v\}$
   c. $FinishTime \leftarrow \max\{FinishTime, StartTime(v) + t_v\}$
   d. מחק את v מהגרף G.
4. החזר את $FinishTime$

---

[1] תמיד קיים קודקוד כזה כי ב-G אין מעגלים – זה נובע מכך שאין סתירה בין התלויות בין המשימות!

הסיבוכיות של האלגוריתם:

- הוראה 1: $O(1)$ פעולות
- הוראה 2: ב-G יש n קודקודים ו-m קשתות. הוספת קודקוד או קשת עולה $O(1)$ ולכן סיבוכיות שלב זה היא $O(n+m)$.
- הוראה 3: הלולאה מתבצעת n פעמים (כי בכל מעבר מורידים קודקוד אחד מ-G ועוצרים כאשר אין יותר קודקודים). מציאת קודקוד ללא קשת שנכנסת אליו (3-a) עולה $O(|V|)=$ $O(n)$ [$|V|$ מתשנה תוך כדי הריצה, אך תמיד קטן מ-n]. הלולאה 3-b עולה בממוצע $O\left(\frac{|E|}{|V|}\right)+O(1)=O\left(\frac{m}{n}+1\right)$ לפי משפט מההרצאה. זה נכון גם עבור 3-d. ההוראה 3-c עולה $O(1)$. לכן, בהוראה 3 מתבצעות $O\left(n\left(n+\left(\frac{m}{n}+1\right)+1+\left(\frac{m}{n}+1\right)\right)\right)=O(n^2+m)$ פעולות.
- הוראה 4: $O(1)$ פעולות.

סה"כ קיבלנו $O(n^2+m+n+m)=O(n^2+m)$ פעולות – פולינומיאלי ב-n,m!

הוכחת נכונות האלגוריתם: נראה שכאשר מוציאים משימה v מתוך G אז ערכו של $StartTime(v)$ הוא הזמן המינימלי בו ניתן להתחיל את v (בהינתן התלויות). אז ינבע ש-$StartTime(v)+t_v$ הוא הזמן המוקדם ביותר בו ניתן לסיים את v. היות ו-$FinishTime$ מכיל בסוף האלגוריתם את המקסימום של כל הגדלים האלה, $FinishTime$ הוא הזמן המוקדם ביותר בו ניתן לסיים את כל המשימות.

נניח ש-v מוצא מ-G. אז אין קודקודים עם קשת אל v ב-G (אחרת לא היינו מוציאים את v ב-3-d). <u>אם</u> v הוא הקודקוד הראשון שיצא מ-G, אז אין משימה שצריכה להתבצע לפני v ואפשר להתחיל אותה בזמן 0 ובאמת $StartTime(v)=0$ (בהכרח זה גם זמן ההתחלה המינימלי). <u>אחרת</u>, כל המשימות שהיו חייבות להתבצע לפני v לפי התלויות כבר הוצאו מ-G. נסמן קבוצה זו ב-U.
באינדוקציה, לכל $u\in U$ מתקיים ש-$StartTime(u)$ הוא הזמן המינימלי בו אפשר להתחיל את u. את v אפשר להתחיל רק לאחר שכל המשימות ב-u נגמרו, כלומר הזמן המינימלי בו אפשר להתחיל את v הוא $\max_{u\in U}(StartTime(u)+t_u)$. הצמתים $u\in V$ עבורם התבצעה הוראה 3-b-i עבור הקשת (u,v) הם בדיוק הקבוצה U (=קבוצת כל הקודקודים מהם הייתה קשת אל v) ולכן כאשר v מוצא מהגרף מתקיים $StartTime(v)=\max_{u\in U}(StartTime(u)+t_u)$, אז גמרנו.

הערה: אפשר לשפר את האלגוריתם כך שירוץ בזמן $O(n+m)$. במקום לחפש כל פעם קודקוד ללא קשתות המובילות אליו, ניתן לבצע מיון טופולוגי ל-G ולעבור על הקודקודים לפי הסדר שהתקבל על הקודקודים.

שאלה 2

נתון גרף מכוון $G=(V,E)$ בו הקשתות צבועות ב-3 צבעים: אדום כחול וירוק.

1. כתבו אלגוריתם המקבל שני קודקודים ומכריע האם יש מסלול בין הקודקודים כך שכל שתי קשתות סמוכות במסלול הן בצבעים שונים.
2. כתבו אלגוריתם המקבל שני קודקודים ומכריע האם יש מסלול בין הקודקודים העובר דרך קשתות מכל הצבעים.

על האלגוריתמים לרוץ ב-$O(|V|+|E|)$ פעולות. הוכיחו את נכונות האלגוריתמים.

פיתרון סעיף 1

אפשר לבצע DFS/BFS מתוחכם בו כל קודקוד יכול להיכנס למחסנית לכל היותר 3 פעמים (רק אם הגענו אליו מקשתות בשלוש צבעים שונים). נתאר כאן פיתרון שקול לפיתרון הזה, אך יותר אלגנטי.

סימון: $C = \{R, G, B\}$ תהייה קבוצת הצבעים של הקשתות.

פיתרון: נגדיר גרף חדש G' באופן הבא: קבוצת הקודקודים תהייה איחוד של שלושה עותקים של V: $V' = V_R \cup V_G \cup V_B$ באשר $V_c = \{v_c | v \in V\}$ לכל $c \in C$ (הסימון v_c הוא סימון פורמלי). הקשתות יוגדרו באופן הבא: לכל $(u,v) \in E$ בצבע c תהייה קשת מ-u_x ל-v_c לכל $c \neq x \in C$. (לדוגמא, אם (u,v) אדומה אז תהיינה קשתות מ-u_B ו-u_G אל v_R.) נסמן את קבוצת הקשתות החדשות ב-E'.
[שימו לב שלפי ההגדרה, קשתות שמובילות ל-v_c ב-G' מגיעות מקשתות בצבע c ב-G.]

נתאר את האלגוריתם (קלט: הגרף G ושני קודקודים u, v):

1. בנה את הגרף $G' = (V', E')$.
2. עבור $c \in \{R, G\}$:
   a. בצע BFS ב-G' שמתחיל מקודקוד u_c. אם הגענו לאחד מהקודקודים v_R, v_B, v_G:
      i. החזר שקיים מסלול כנדרש
3. החזר שאין מסלול כנדרש.

בגרף G' יש $3|V|$ קודקודים ו-$2|E|$ קשתות ולכן בנייתו דורשת $O(|V| + |E|)$ פעולות. מאותה סיבה, ביצוע BFS על G' עולה $O(3|V| + 2|E|) = O(|V| + |E|)$ פעולות. אנו עושים 2 BFS-ים כאלה ועוד מספר קבוע של פעולות ולכן סך כל העבודה היא $O(|V| + |E|)$.

האלגוריתם מחזיר שיש מסלול כנדרש אם"ם יש מסלול בין אחד מהקודקודים $\{u_R, u_G\}$ אל אחד מהקודקודים $\{v_R, v_G, v_B\}$. לכן, <u>כדי להוכיח נכונות</u> מספיק להראות שהנ"ל מתקיים אם"ם קיים מסלול מ-u ל-v בו קשתות סמוכות הן בצבעים שונים.

<u>נניח ש-$u = u^{(0)} \to u^{(1)} \to \cdots \to u^{(t)} = v$ מסלול ב-G</u> בו כל שתי קשתות סמוכות בצבעים שונים. נסמן לכל $i > 1$: $c_i = Color(u^{(i-1)}, u^{(i)})$, אזי לפי הגדרת E' יש קשת מ-$u^{(i)}_{c_i}$ ל-$u^{(i+1)}_{c_{i+1}}$ ב-G' (כי $c_{i+1} \neq c_i$) לכל $1 < i < t$. בנוסף, יש קשת מ-$u^{(0)}_c$ אל $u^{(1)}_{c_1}$ לכל $c \neq c_1$. תמיד קיים $c \in \{R, G\}$ כך ש-$c \neq c_1$ ולכן קיים מסלול ב-G' מאחד מהקודקודים $\{u_R, u_G\}$ אל אחד מהקודקודים $\{v_R, v_G, v_B\}$ (המסלול הוא $u_c = u^{(0)}_c \to u^{(1)}_{c_1} \to \cdots \to u^{(t)}_{c_t} = v_{c_t}$).

<u>נניח שיש מסלול ב-G':</u> $u_{c_0} = u^{(0)}_{c_0} \to u^{(1)}_{c_1} \to \cdots \to u^{(t)}_{c_t} = v_{c_t}$. אזי מהגדרת E' נובע שלכל $0 \leq i < t$ מתקיים $(u^{(i)}, u^{(i+1)}) \in E$, $Color(u^{(i)}, u^{(i+1)}) = c_{i+1}$ ו-$c_i \neq c_{i+1}$. לכן, $u = u^{(0)} \to u^{(1)} \to \cdots \to u^{(t)} = v$ הוא מסלול ב-G בו קשתות סמוכות הן בצבעים שונים.

הערה: ניתן להכליל את הפיתרון ל-k צבעים כך שיעבוד ב-$O\big(k(|V| + |E|)\big)$ פעולות.

פיתרון סעיף 2

שוב, במקום BFS/DFS מתוחכם, נמיר את הבעיה לבעיה בגרף אחר.

סימון: $C = \{R, G, B\}$ תהייה קבוצת הצבעים של הקשתות.

נגדיר גרף חדש $G' = (V', E')$: הקבוצה V' תורכב מ-8 עותקים של V: $V' = \cup_{X \subseteq C} V_X$ באשר $V_X = \{v_X | v \in V\}$ (הסימון v_X הוא פורמלי). הקשתות יוגדרו באופן הבא: אם $(u,v) \in E$ בצבע c ו-$X \subseteq C$, אז $(u_X, v_{X \cup \{c\}}) \in E'$. [אינטואיציה: האינדקס X ב-v_X זוכר באילו צבעים ביקרנו בדרך ל-v.]

האלגוריתם (קלט: הגרף G ושני קודקודים u, v):

1. בנה את G' מ-G.
2. בצע BFS ב-G' מהקודקוד u_ϕ. אם הגענו אל $v_{\{R,G,B\}}$ החזר שיש מסלול כדרוש, אחרת החזר שאין מסלול כנדרש.

בגרף G' יש $8|V|$ קודקודים ו-$8|E|$ קשתות ולכן בנייתו דורשת $O(|V| + |E|)$ פעולות. מאותה סיבה, ביצוע BFS על G' עולה $O(8|V| + 8|E|) = O(|V| + |E|)$ פעולות. סה"כ קיבלנו שהאלגוריתם מבצע $O(|V| + |E|)$ פעולות.

כדי להוכיח שהאלגוריתם עובד צריך להראות שקיים מסלול מ-u ל-v העובר דרך קשתות מכל הצבעים *אם"ם* קיים ב-G' מסלול מ-u_ϕ ל-$v_{\{R,G,B\}}$.

<u>נניח שקיים מסלול $u = u^{(0)} \to u^{(1)} \to \cdots \to u^{(t)} = v$ העובר בקשתות בכל הצבעים</u>. נסמן $c_i = Color\left(u^{(i-1)}, u^{(i)}\right)$ ונגדיר $X_i = \{c_1, c_2, \dots, c_i\} = X_{i-1} \cup \{c_i\}$ $(X_0 = \phi)$. אזי $X_t = \{R, G, B\}$. לפי הגדרת E', $\left(u_{X_{i-1}}^{(i-1)}, u_{X_i}^{(i)}\right) \in E'$ לכל $0 < i \leq t$ ולכן יש מסלול מ-$u_\phi = u_{X_0}^{(0)}$ אל $v_{\{R,G,B\}} = u_{\{R,G,B\}}^{(t)} = u_{X_t}^{(t)}$, כדרוש.

<u>נניח שיש מסלול</u> $u_\phi = u_{X_0}^{(0)} \to u_{X_1}^{(1)} \to \cdots \to u_{X_t}^{(t)} = v_{\{R,G,B\}}$ ב-G'. אזי לכל $0 \leq i < t$ מתקיים $\left(u^{(i)}, u^{(i+1)}\right) \in E$ ו-$X_{i+1} = X_i \cup \left\{Color\left(u^{(i)}, u^{(i+1)}\right)\right\}$. לכן ב-$G$ קיים המסלול, $u = u^{(0)} \to u^{(1)} \to \cdots \to u^{(t)} = v$. נראה שהוא עובר בקשתות מכל הצבעים. נסמן $c_i = Color\left(u^{(i)}, u^{(i+1)}\right)$. אזי $\{R, G, B\} = X_t = X_{t-1} \cup \{c_{i-1}\} = \cdots = X_0 \cup \{c_0, c_1, \dots, c_{i-1}\} = \{c_0, c_1, \dots, c_{i-1}\}$ ולכן לכל $c \in \{R, G, B\}$ קיים $0 \leq i < t$ כך ש-$c = c_i = Color\left(u^{(i)}, u^{(i+1)}\right)$. כדרוש.

הערה: אפשר להכליל את הפיתרון ל-k צבעים ואז הסיבוכיות תהייה $O\left(2^k(|V| + |E|)\right)$.

שאלה 3

נתון גרף קשיר לא מכוון G=(V, E) ממושקל, ועץ פורש מינימלי T בתוכו. מוסיפים קשת (u, v) חדשה ל-G. מצא עץ פורש מינימלי T' בגרף $G' = G \cup \{(u, v)\}$, בזמן O(n).

<u>**פתרון:**</u>

תאור האלגוריתם:

1. נריץ את אלגוריתם DFS* מ-u על $T \cup \{(u, v)\}$.

DFS* דומה לאלגוריתם DFS "הרגילי", פרט לשינויים הבאים:

<u>DFS(G, s)</u>*

1. for each vertex w∈V[G]
2. color[w] ← WHITE
3. π[w] ← NIL

```
time ← 0
DFS-VISIT*(s)
```

<u>DFS-VISIT(w)</u>*

```
color[w] ← GRAY
d[w] ← time ← time + 1
for each z∈Adj[w]
    if ( z = s )
        stop
    if ( color[z] = WHITE )
        π[z] ← w
        DFS-VISIT*(z)
color[w] ← BLACK
f[w] ← time ← time + 1
```

2. נריץ את אלגוריתם FindMax(w) : (w הוא הקודקוד ש״גילה מחדש״ את u)

<u>FindMax(s)</u>

```
max ← 0
while (s ≠ u)
    if (max < w(s, π[s]))
        max ← w(s, π[s])
    s ← π[s]
```

3. אם $max \le w(u, v)$, אזי T' הוא T.

אם $max > w(u, v)$, אזי נחליף את הקשת בעלת המשקל max עם הקשת (u, v) ב-T, ונקבל את T'.

<u>סיבוכיות:</u>

1. ברור שהסיבוכיות זהה לסיבוכיות של DFS ״הרגיל״ : $O(m + n)$.
2. ריצה על הקדקודים על-פי האב : $O(n)$.
3. בדיקה : $O(1)$.

סה״כ: $O(m + n)$.

<u>נכונות:</u>

ברור שהקשת (u, v) סוגרת מעגל ב-T ⇐ אם קיימת קשת e במעגל זה בעלת משקל הגדול מ-w(u, v), האלגוריתם יחליף בין e ו-(u, v). כך יתקבל עץ 'T שמשקלו קטן ממשקל T.

נותר להוכיח שאין קשת השייכת ל-G\T' הניתנת להחלפה בקשת השייכת ל-'T.

נניח שאכן בוצעה ההחלפה בסעיף 3. אזי: $G\setminus T' = (G\setminus T) \cup \{e\}$. ⇐ די להוכיח שלא קיימת קשת השייכת ל-G\T אשר אם נחליפה בקשת השייכת ל-'T, נקבל עץ ''T בעל משקל קטן יותר.

כל קשת השייכת ל-G\T סוגרת מעגל ב-'T (וכן סוגרת מעגל ב-T). תהי קשת e=(i, j) השייכת ל-G\T. אם e סוגרת מעגל ב-'T שאינו מכיל את הקשת (u, v), אזי ברור שלא ניתן להחליף את e עם קשת השייכת ל-'T (מכיוון ש-T הוא עץ פורש מינימלי).

אם e סוגרת מעגל ב-'T המכיל את הקשת (u, v), אזי e סוגרת מעגל ב-T, המכיל את המסלול מ-u ל-v.

⇐ $w(e) \le$ ממשקל כל קשת במסלול מ-i ל-j ב-T. (אחרת e הייתה שייכת ל-T). כמו כן, $w(u, v) >$ מהמשקל המקסימלי של הקשתות במסלול "העוקף" מ-u ל-v ב-T (על פי סעיף 3).

⇐ $w(e) \le$ ממשקל של כל קשת במסלול מ-i ל-j ב-'T.

⇐ אם נחליף את e בקשת אחרת השייכת למסלול מ-i ל-j, *<u>לא</u>* נפחית ממשקלו של 'T.