אלגוריתמים בתורת הגרפים –פתרון תרגיל מס' 2

לשאלות והערות נא לפנות לאילן גרונאו (shrilan@cs.technion.ac.il)

1) *גרף **דו-צדדי** (bipartite) הינו גרף $G(V,E)$ אשר קיימת חלוקה של צמתיו לשתי קבוצות U,W ($U \cup W = V, U \cap W = \phi$) כך שכל קשת $e \in E$ הינה מצורת $e=(u,w)$ או $e=(w,u)$ כאשר $u \in U, w \in W$.*

   א) *מצא תנאי מספיק והכרחי לכך שגרף לא-מכוון הוא דו-צדדי.*
   ב) *הראה אלגוריתם <u>יעיל</u> שבודק אם גרף הוא דו-צדדי ואם כן מציע חלוקה U,W מתאימה.*
   ג) *נתח את סיבוכיות האלגוריתם שהצעת.*

פתרון שאלה 1:

1. גרף לא מכוון $G(V,E)$ הוא דו צדדי אס"ם לא קיים בו מעגל באורך אי זוגי.

<u>התנאי הכרחי</u>:

נניח $G(V,E)$ גרף דו צדדי, שכל צמת בו שייך או ל U או ל W (באופן בלעדי). נניח שיש בו מעגל באורך אי זוגי (לאו דווקא פשוט) $(v_1, v_2, \cdots v_{2k+1}, v_1)$. נסמן ב U (בלי הגבלת הכלליות) את הקבוצה בה נמצא v_1. ניתן לראות (באינדוקציה פשוטה) כי כל צמת v_i עבור i זוגי שייך ל W, וכל צמת v_i עבור i אי זוגי (בפרט v_{2k+1}) שייך ל U. מכאן ש $v_1, v_{2k+1} \in U$. אבל $(v_1, v_{2k+1}) \in E$ בסתירה להנחה על דו צדדיות הגרף.

<u>התנאי מספיק</u>:

בסעיף הבא נראה שאם אין בגרף מעגל באורך אי זוגי האלגוריתם המוצע מוצא חלוקה של צמתי הגרף לקבוצות זרות ומשלימות. לכן התנאי מספיק למציאת חלוקה מתאימה לשתי קבוצות.

2. אלגוריתם לחישוב ובדיקת גרף דו צדדי בהינתן גרף לא מכוון $G(V,E)$, ושתי קבוצות ריקות U,W. נבצע אלגוריתם שהוא ואריאציה על BFS מצמת שרירותי s בגרף. כל צמת במרחק זוגי נכניס ל U וצמת במרחק אי זוגי נכניס ל W:

<u>אתחול</u>:

נאתחל תור ריק Q, לכל $v \in V$ $d[v] = \infty$, ו- $U = W = \phi$.

כל עוד קיים צמת s ב V כך ש $d[s] = \infty$:

הכנס את s לראש התור ואתחל $d[s] = 0$.

<u>כל עוד Q אינו ריק</u>:

הוצא את v מראש התור. אם $d[v]$ זוגי הכנס את v ל-U, אחרת הכנס אותו ל-W.

לכל צמת u שכן של v בצע:

- אם $d[u] = \infty$ אז הכנס את u לסוף התור ו- $d[u] := d[v]+1$.
- אחרת אם $d[v]+ d[u]$ זוגי עצור ופלוט: "הגרף הוא לא דו צדדי!".

הסבר: כמו BFS האלגוריתם מעדכן פעם אחת בלבד את $d[v]$ לכל צמת בגרף, ומוודא לכל קשת (u,v) פעם אחת לפחות כי $d[v]+ d[u]$ אי זוגי. $d[v]+ d[u]$ אי זוגי אס"ם הזוגיות של v ו-u שונה, כלומר הם נמצאים בקבוצות שונות. לכן אם האלגוריתם פולט חלוקה לשתי קבוצות אז הגרף דו צדדי. אם הוא פולט "הגרף הוא לא דו צדדי!" אז קיימת קשת (u,v) בגרף כך שסכום המרחקים בין צמת s כלשהו בגרף ל-v ול-u הוא זוגי. קיים, אם כן, מעגל (לאו דווקא פשוט) s-…-u-v-…s באורך אי זוגי לכן הגרף $G(V,E)$ אכן אינו גרף דו צדדי. אם לא קיים מעגל אי זוגי בגרף האלגוריתם לא ייעצר לפני שכיסה את כל צמתי הגרף.

הערות:

- ניתן גם לבצע אלגוריתם דומה כואריאציה על DFS, אך מתבקש במקרה זה להשתמש ב BFS כיוון שניתן להסתמך על הנכונות, ולהיעזר במרחקים שהוא מחשב.
- חשוב לוודא שכיסינו את כל צמתי הגרף, ולא להפסיק כשהתור Q מתרוקן.

3. **סיבוכיות:** O(|E|+|V|). האלגוריתם פועל כמו BFS, כך שבכל איטרציה מוסיפים מספר קבוע של פעולות (בדיקות והוספה לקבוצות U ו W).

2) ***קוטר*** *הגרף הינו המקסימום מבין המרחקים המינימליים בין זוגות צמתים בגרף.*
א) *מצא אלגוריתם יעיל לחישוב הקוטר של עץ לא-מכוון.*
ב) *נתח את סיבוכיות האלגוריתם שהצעת.*

פתרון שאלה 2:

1.
הרעיון: נבצע שתי ריצות BFS. האחת מצמת שרירותי s בגרף, והשניה מצמת u הרחוק ביותר מ-s. קוטר הגרף הוא המרחק בין u לצמת הרחוק ביותר ממנו- v.

האלגוריתם:

- בחר צמת שרירותי s בעץ T, ובצע אלגוריתם BFS סטנדרטי על העץ מ- s.
- יהי u הצמת שיצא אחרון מתור BFS בשלב הקודם.
- בצע אלגוריתם BFS מצמת u. יהי v הצמת שיצא אחרון מתור BFS של u.
- קוטר העץ הוא המרחק של v מ-u.

הסבר: המרחק בין שני צמתים בעץ-T הוא אורך המסלול הפשוט <u>היחיד</u> ביניהם. בנוסף ניעזר בטענה הבאה: (*קלה להוכחה*)
בהינתן עץ BFS מכוון המושרה על T, ומסלול פשוט כלשהו- $u_0 - u_1 - \cdots - u_k$ קיים צמת u_i (שנקרא לו "ראש המסלול") במסלול כך שלכל $0 < j \le i$ מתקיים u_j אבא של u_{j-1}, ולכל $i < j \le k$ מתקיים u_{j-1} אבא של u_j.
* נשים לב שראש המסלול הוא אב קדמון של כל הצמתים במסלול.

נראה כי כל מסלול פשוט (מרחק) בגרף אינו ארוך יותר מהמסלול הפשוט בין v ל-u (שנסמן ב-p). כלומר שהמרחק בין v ל-u הוא מירבי. יהיו x,y שני צמתים כלשהם בגרף ו p_1 המסלול הפשוט ביניהם, שהראש שלו (ביחס לעץ BFS המכוון ש-s הוא שרשו) הוא w_1.
נתבונן במסלול הפשוט p_2 מ-u ל-y. נסמן ב-w_2 את ראש המסלול p_2, וב-z את הצמת הראשון המשותף ל-p_1 ו-p_2. כך שמתקיים $p_1 = x \xrightarrow{q_1} z \xrightarrow{q} y$ ו $p_2 = u \xrightarrow{q_2} z \xrightarrow{q} y$ (*מדוע חלוקה זו אפשרית?*). בנוסף מתקיים w_2 אב קדמון או צאצא של w_1 (*מדוע זה מתחייב?*), לכן נוכל לסמן את ה"בוגר" מבין השניים ב-w. w הוא אב קדמון של x,y,u (*מדוע?*).

- <u>נניח הצמת w_1 נמצא ב-q</u>. כלומר ש-w נמצא ב-p_2 ובנוסף הוא ראש המסלול (*מדוע?*). נוכל אם כן לסמן מחדש: $p_2 = u \xrightarrow{r} w \xrightarrow{r'} y$. מנכונות BFS נובע כי u הוא צמת בעל מרחק מירבי מ- s, ו v בעל מרחק מירבי מ-u. לכן מתקיימת סדרת אי השוויונים (*מדוע?*):

$$d(u,v) \ge d(u,y) = |p_2| = |r| + |r'| = d(w,u) + d(w,y) \ge d(w,x) + d(w,y) \ge d(w_1,x) + d(w_1,y) = |p_1|$$

- <u>מה קורה אם w_1 נמצא ב-q_1</u>?

2. **סיבוכיות:** האלגוריתם הוא על בסיס הפעלה כפולה של אלגוריתם BFS עם תוספת קבועה של פעולות בדיקה. מכאן שסיבוכיות האלגוריתם המוצע היא $O(|E|+|V|) = O(|V|)$, כיוון שבעץ לא מכוון מתקיים $|E| = |V|-1$.
הערה: רצוי להביא את הביטוי לסיבוכיות לצורה הפשוטה ביותר שלו כפונקציה של מספר הצמתים (אם אפשר) ומספר הקשתות (אם צריך).

3) *נתון גרף $G(V,E)$ אשר כל אחת מקשתותיו צבועה באדום או שחור, וצומת $s \in V$.*

א) *הראה אלגוריתם, שיעילותו $O(|E|+|V|)$, המוצא לכל צמת v בגרף את המסלול הקצר ביותר מ- s העובר בקשת אדומה אחת <u>לפחות</u>.*

ב) *הראה אלגוריתם, שיעילותו $O(|E|+|V|)$, המוצא לכל צמת v בגרף את המסלול הקצר ביותר מ- s העובר בקשת אדומה אחת <u>בדיוק</u>.*

<u>רמז</u>: השתמשו ברדוקציה ל-BFS. (רדוקציה הינה טרנספורמציה של בעיה נתונה לבעיה אחרת, שאת פתרונה ניתן לתרגם לפתרון הבעיה המקורית).

פתרון שאלה 3:

<u>פתרון א'</u>: ניתן לבצע וריאציה על BFS כאשר לכל צמת *v* מעדכנים שני ערכים תוך כדי הריצה:
$\lambda_b(v)$: אורך המסלול השחור הקצר ביותר מ-*s* ל-*v*. $\lambda(v)$: אורך המסלול הקצר ביותר מ-*s* ל-*v* המכיל קשת אדומה אחת לפחות/בדיוק. האלגוריתם דומה מאוד לאלגוריתם לחישוב אורך מסלול זוגי מינימלי שהוצג בתרגול. יש לשים לב שכל צמת יכול להיכנס לתור ה-BFS (להתעדכן) פעמיים (לכל היותר).

<u>פתרון ב'</u>: **רדוקציה ל-BFS**.

1.

נעתיק את הבעיה לבעיה שקולה (על גרף אחר) הניתנת לפתרון על ידי אלגוריתם BFS. יש לוודא שההעתקה לא מוסיפה לסיבוכיות של אלגוריתם BFS. בהינתן גרף $G(V,E)$ נגדיר גרף שכבות $\overline{G}(\overline{V},\overline{E})$ כך ש- $\overline{V}=V\cup V'(V'=\{v' \mid v\in V\})$ ו- $\overline{E}=E\cup E'\cup E_{red}$ כך ש:
$(E'=\{(u',v') \mid (u,v)\in E\}, E_{red}=\{(u',v) \mid (u,v) \text{ is a red edge in } G\})$.

נריץ אלגוריתם BFS על הגרף החדש $\overline{G}(\overline{V},\overline{E})$ מצמת *s'*. נוכיח כי לכל צמת *v* ב-*V* מתקיים $d(s',v)$ (המרחק בין שני הצמתים ב-$\overline{G}$) הוא אורך המסלול הקצר ביותר מ- *s* ל- *v* העובר בקשת אדומה אחת לפחות.

- כל מסלול מ-*s'* ל- *v* עובר בקשת השייכת ל- E_{red} (משפט "מסלול חוצה חתך").
- $d(s',v)$ הוא אורך המסלול הקצר ביותר מ-*s'* ל- *v*. מסלול זה נראה כך: $\overline{p}=(s'-\cdots-u'-w-\cdots-v)$ כאשר (u',w) קשת ב- E_{red}. לכן המסלול $p=(s-\cdots-u-w-\cdots-v)$ הנמצא ב-*G*, העובר בקשת אדומה אחת לפחות- (u,w). מכאן שקיים מסלול באורך $d(s',v)$ העובר בקשת אדומה אחת לפחות.
- יהי $p=(s-u_1\cdots-u_i-u_{i+1}-\cdots-v)$ מסלול כלשהו ב-*G*, <u>העובר בקשת אדומה אחת לפחות</u>- (u_i-u_{i+1}). אז המסלול $\overline{p}=(s'-u_1'\cdots-u_i'-u_{i+1}-\cdots-v)$ נמצא ב-$\overline{G}$ ואורכו זהה לזה של *p*. לכן מנכונות BFS נובע: $|p|=|\overline{p}|\le d(s',v)$. **<u>מ.ש.ל</u>**

2. נבצע העתקה דומה. הפעם לגרף **<u>מכוון</u>**: בהינתן גרף $G(V,E)$ נגדיר גרף שכבות **<u>מכוון</u>** $\overline{G}(\overline{V},\overline{E})$ כך ש- $\overline{V}=V\cup V'(V'=\{v' \mid v\in V\})$ ו- $\overline{E}=E^1_{black}\cup E^2_{black}\cup E_{red}$
$E^1_{black}=\{(u,v) \mid (u,v) \text{ is a black edge in } G\}, E^2_{black}=\{(u',v') \mid (u,v) \text{ is a black edge in } G\}$
$E_{red}=\{(u',v) \mid (u,v) \text{ is a red edge in } G\}$.
ניתן להראות נכונות באופן דומה. ההבדל כאן הוא שבגרף הנוצר ע"י הרדוקציה הזו בכל מסלול מכוון יש לכל היותר קשת אדומה אחת (כי לא ניתן לעבור מ-*V* ל-*V'*).

4) **שורש** *בגרף מכוון הינו צומת בגרף ממנו קיים מסלול מכוון לכל הצמתים בגרף.*

א) *נתון גרף מכוון* $G(V,E)$. *הצע אלגוריתם שסיבוכיותו* $O(|E|)$ *המוצא שורש ב-* G*, או מודיע שאין כזה.*

ב) *הוכח את נכונות האלגוריתם.*

פתרון שאלה 4:

1.

הרעיון: נריץ DFS מצמת שרירותי s בגרף. אם פרשנו את כל הגרף אז s הוא שורש של הגרף. אם לא- נבחר צמת אחר אליו טרם הגענו ונריץ ממנו DFS, וחוזר חלילה עד שנכסה את כל הגרף ע"י יער DFS. יהי r הצמת האחרון ממנו הרצנו DFS; הוא "מועמד לשרש הגרף". נריץ ממנו DFS (על כל הגרף מחדש). אם עץ ה-DFS החדש פורש את הגרף אז r הוא שרש הגרף. אחרת לא קיים שרש לגרף.

האלגוריתם:

<u>מציאת "מועמד לשרש הגרף"</u>:

סמן כל v ב-V כ"*חדש*". אתחל מחסנית S ריקה.

כל עוד קיים s "*חדש*" ב-V:

- סמן את s כ"*ישן*" והכנס אותו למחסנית S.
- כל עוד המחסנית S אינה ריקה:
  - הוצא את v מראש המחסנית.
  - הכנס כל שכן "*חדש*" של v לראש המחסנית.

<u>בדיקת המועמד</u>:

יהי r הצמת האחרון שהוצאנו מהמחסנית. סמן את כל הצמתים ב-V כ"*חדשים*" ובצע DFS מ-r.
אם כל הצמתים "*ישנים*" החזר את r בתור שרש. אם לא החזר: "*אין שרש בגרף* $G(V,E)$".

סיבוכיות: אנו מבצעים שתי סריקות אתחול לצמתים- $O(|V|)$, ושתי סריקות על כל הקשתות בגרף (אחת בבניית יער ה-DFS, ואחת באלגוריתם הבדיקה בסוף)- $O(|E|)$. לפני ריצת האלגוריתם נבדוק כי $|E|>|V|-2$. אם התנאי לא מתקיים אז הגרף לא קשיר (*מדוע?*) ולגרף אין שרש. אם $|E|>|V|-2$ נריץ את האלגוריתם למעלה בסיבוכיות $O(|E|+|V|)=O(|E|)$.

2. הוכחת נכונות:

I. אם האלגוריתם למעלה מחזיר צמת r בתור שרש אז r שרש של G.

אחרי ריצת DFS בסוף האלגוריתם כל צמת בגרף שייך לעץ DFS ששורשו r, לכן מנכונות DFS נובע כי קיים מסלול מכוון מ-r לכל צמת בגרף G.

II. אם קיים שרש s בגרף מכוון G אז האלגוריתם למעלה מחזיר צמת כלשהו r שהוא שרש של G (האם בהכרח $r=s$?).

מנכונות DFS, בו השתמשנו בחלק הראשון של האלגוריתם (מציאת "מועמד לשרש הגרף") כל צמת בגרף נכנס פעם אחת בדיוק למחסנית ויוצא ממנה פעם אחת בדיוק. נתבונן בצמת r הנמצא בתחתית המחסנית ברגע בו הוצאנו את השרש s מהמחסנית.

טענה 1: <u>r שרש של הגרף G</u>. מנכונות DFS נובע כי בכל רגע נתון קיים מסלול (מכוון) מהצומת בתחתית המחסנית (שרש עץ ביער DFS של הגרף G) לכל הצמתים הנמצאים באותו רגע במחסנית; לכן s ישיג מ-r. כל צמת בגרף ישיג מ-s (כי הוא שרש של G), ולכן כל צמת בגרף ישיג גם מ-r.

טענה 2: <u>אחרי הוצאת הצמת r מהמחסנית כל הצמתים בגרף מסומנים "ישנים"</u>.

כל הדיון הבא מתייחס לנקודה בזמן בה הוצאנו את r מהמחסנית. נניח קיים צמת v שאינו מסומן כ"*ישן*" (לא הוכנס ל-S). v ישיג מ-r (טענה 1), לכן קיים מסלול מכוון $p=(r-u_1\cdots u_i-u_{i+1}\cdots v)$.

יהי u_i הצמת האחרון במסלול p (בעל אינדקס מירבי) שאינו מסומן כ"*ישן*" (קיים כזה מפני ש-r מסומן כ"*ישן*" ו-v לא). u_{i+1} אינו מסומן כ"*ישן*". u_i הוא "*ישן*" אך אינו נמצא במחסנית (הוצאת r

מהמחסנית משאירה אותה ריקה), לכן הוא יצא מהמחסנית. גם ברגע הוצאתו של u_i מהמחסנית u_{i+1} לא היה מסומן כ"ישן" (צמת המסומן כ"ישן" נשאר "ישן"). מיד אחרי הוצאתו של u_i מהמחסנית מכניסים את כל השכנים ה"חדשים" שלו (בפרט u_{i+1}) למחסנית ומסמנים אותם כ"ישנים", בסתירה לכך ש-u_{i+1} עדיין אינו מסומן כ"ישן".

מסקנה: אחרי הוצאת r מהמחסנית כל הצמתים "ישנים", לכן r הוא "מועמד לשרש הגרף". הצמת r הוא אכן שרש של הגרף G, ולכן ריצת הבדיקה תצליח ו-r יוחזר כשרש הגרף. לכן האלגוריתם מחזיר צמת כלשהו (r) שהוא שרש של G.

מ.ש.ל

5) ***רכיב קשיר-היטב*** *בגרף מכוון* $G(V,E)$ *הינו תת-קבוצה של צמתים* $V' \in V$ *כך שלכל זוג צמתים ברכיב* $u,v \in V'$ *קיימים ב-*G *מסלולים מכוונים* $u \to v\,,\, v \to u$.

א) מצא אלגוריתם למציאת רכיבים קשירים-היטב בגרף מכוון, שיעילותו $O(|E|+|V|)$.

פתרון שאלה 5:

נגדיר את הגרף ההפוך ל-G: $G^T(V,E^T)$ כך ש-$E^T = \{(u,v) \mid (v,u) \in E\}$.
ניעזר בטענות הבאות (מומלץ לנסות להוכיח עצמאית):

1. רכיבי הצמתים הקשירים היטב (מקסימליים) ב- G וב-G^T - זהים.
2. בכל ריצת DFS על G מקבלים יער של עצי-DFS. כל רכיב קשירות מוכל (על כל צמתיו) בעץ DFS כלשהו.
3. בהינתן יער DFS של הגרף G ניתן לסווג כל קשת (u,v) בגרף לאחת הקבוצות הבאות לפחות:
   - u,v באותו רכיב קשירות.
   - finish[v]<finish[u], כאשר finish[v] מציין את זמן הנסיגה מ-v בהרצת DFS.
4. מסקנה: אין קשתות נכנסות לרכיב הקשירות של הצמת ממנו DFS נסוג אחרון. כלומר שב-G^T אין קשתות <u>יוצאות</u> מרכיב הקשירות של הצמת ממנו DFS נסוג אחרון.

האלגוריתם:

- הרץ DFS על הגרף G על מנת לקבל יער DFS מכוון הכולל את כל צמתי הגרף. בכל פעם שהשגרה הרקורסיבית נסוגה מצמת v הכנס אותו למחסנית S.
- חשב את הגרף ההפוך-G^T.
- הרץ DFS על הגרף G^T על מנת לקבל יער DFS, כאשר סדר בחירת הצמתים (שרשי עצי היער) הוא בהתאם לסדר הופעתם במחסנית S (סדר יורד של finish). כל עץ DFS ביער המתקבל הוא רכיב קשיר היטב (מקסימלי) של G.

נכונות: נובעת מהטענות למעלה + אינדוקציה על מספר רכיבי הקשירות המקסימליים ב-G.

סיבוכיות: שתי ריצות DFS וחישוב G^T: $O(|E|+|V|)$.