פתרון ממ"ן 11 בקורס אלגוריתמים – 2012ב

הפתרון נכתב על ידי הסטודנט שלומי וקנין

שאלה 1

א. נתון עץ T בעל דרגה 3 לכל היותר.

טענה 1: ניתן להסתכל על עץ כזה כעץ בינארי מושרש.

הוכחה: יהי T עץ בעל n צמתים בו לכל צומת דרגה 3 לכל היותר:

1. מהנתון כי T עץ, אנו יודעים כי יש לו n-1 קשתות (3.1) וכן הוא קשיר ואינו מכיל מעגל (3.2).
2. נבחר צומת $e \in T$, נבצע עליו BFS, נקבל עץ סריקה `T, אשר מכיל גם הוא n צמתים, כי T קשיר.
3. מכך של- `T ול T אותו מספר צמתים, הרי שכל $t \in T'$ מתקיים גם $t \in T$
4. וכן, מכך ש `T עץ, יש לו עלה.
5. נבחר עלה $t \in T'$ ולכן גם $t \in T$, ונעשה BFS על t ב T, ונקבל שוב עץ סריקה ``T המושרש ב t.

נראה כי ``$T$ הוא עץ בינארי מושרש: עץ `T מכיל $n$ צמתים ומכאן גם $n$-$1$ קשתות, ולכן כל קשת ב T קיימת גם ב `T, וכן בגלל ש $t$ הוא עלה של `T, הרי של $t$ יש דרגה 1 בדיוק. כלומר מצאנו כי שורש העץ ``T הוא בעל דרגה 1, ולכן יש לו בן אחד בלבד. מהנתון כי דרגת כל שאר הצמתים לא עולה על 3, נובע כי לכל צומת ב ``T שאינה $t$ יש אב אחד (בהכרח) ולא יותר מ-2 בנים, ולכן הראנו כי ``T הוא עץ בינארי המושרש ב t.

עתה נבנה אלגוריתם המוצא קשת כזו בתוך עץ ``T שהוא ייצוג של עץ T:

1. נתחיל בשורש, ונבדוק את כמות הצמתים בכל אחד משני תת העצים (אם אין תת עץ נחזיר 0).
2. נבחר בתת העץ עם מקסימום צמתים.
3. אם בתת העץ שבחרנו יש פחות מ ¾ צמתים מהעץ המקורי,
4. אז מצאנו את הקשת משורש תת העץ שלנו לאביו, נחזיר אותו.
5. אחרת, קרא ברקורסיה עם תת העץ שבחרנו.

נראה נכונות: נסמן: $t = ¾n, n = |T|$

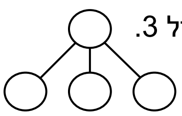
נראה כי האלגוריתם מסתיים:
t הוא מספר קבוע. בכל איטרציה אנו מתחילים עם עץ מסוים, בוחרים תת עץ ומפעילים את הרקורסיה עליו. מכך שאנו מפעילים את הרקורסיה על תת עץ, הרי שמספר הצמתים בתת עץ חייב להיות קטן בלפחות אחד (עבור השורש) ממספר הצמתים בעץ שמכיל אותו. לכן בשלב מסוים המספר הזה בהכרח יקטן מהקבוע t, והאלגוריתם יסתיים.

טענה 2: בסיום האלגוריתם מצאנו שני תתי עצים המכילים כל אחד פחות מ t צמתים

הוכחה: האלגוריתם עוצר <u>בפעם הראשונה</u> בא הוא מוצא תת עץ בעל מספר רב ביותר של צמתים,
המכיל פחות מ t צמתים. נסמן את תת העץ הנ"ל ב G, ונגדיר $s = |G|$, כלומר מתקיים $s < t = ¾n$
אולם, בגלל שזו הפעם הראשונה, בהכרח מתקיים כי $|Parent[G]| \geq ¾n$ (אחרת היינו עוצרים שם),
מכך שאנו בוחרים את תת העץ עם מקסימום צמתים, לא יתכן כי נבחר בתת עץ המכיל פחות מחצי
מכמות הצמתים של האב, ולכן נקבל $\frac{1}{2} * \frac{3}{4}n = \frac{3}{8}n \leq s < \frac{3}{4}n$
כלומר מצאנו עץ אחד בעל פחות מ t צמתים.
נראה כי גם העץ הנוסף מכיל פחות מ t צמתים:
העץ הנוסף יכיל $n - s$ צמתים, כלומר $n - s \leq n - \frac{3}{8}n = \frac{5}{8}n < \frac{6}{8}n = \frac{3}{4}n = t$
והראנו כי גם העץ השני מכיל פחות מ t צמתים, ובכך הוכחנו את הטענה.

כנדרש.

ב. להלן עץ פשוט המקיים את הדרוש:
מתקיים $\frac{3}{4} * 4 = 3$ ולכן כל קשת שנוריד תיתן שני תתי עצים, אחד בגודל 1 והשני בגודל 3.



כנדרש.

<u>שאלה 2</u>

הרעיון הוא להחליף את כל קבוצה *S* בצומת בודדת אחת, ואז נקבל רדוקציה ל *BFS* פשוט.

נתון גרף לא מכוון קשיר $G = (V, E)$

טענה 1: החלפה של קבוצה *S* בצומת בודד השומר על כל קשת שצידה האחד ב *S* וצידה האחר מחוץ ל *S*, לא משנה את המרחק בין הקבוצה לכל שאר הצמתים.

הוכחה: יהי $v, u \in V/S$, ואיבר $s \in S$ המקיימים, קיימת קשת $a = (s, v)$ ו $b = (v, u)$, אך לא קיימת קשת (s, u).

מכאן שמתקיים $d(s, u) \;=\; d(s, v) + d(v, u)$

אולם מקיום *a* נקבל $d(s, u) = d(v, u) + 1$

ניצור צומת חדשה *n*, ניתן לה קשת $a` = (n, v)$,

ומתקיים $d(n, u) = d(n, v) + d(v, u) = 1 + d(v, u) = d(s, u)$

בעזרת טענה 1 נבנה את האלגוריתם הדרוש,

1. ראשית, נוסיף לכל הקשתות שדה צבע, ונצבע אותן בירוק.
2. ניצור צומת חדשה *n*
3. עבור כל קשת $e = (s, v), s \in S, v \in V/S$
4. צבע את *e* בשחור
5. הוסף את הקשת *(n,v)* לגרף, וצבע אותה בסגול
6. בצע *BFS* מותאם אשר מתחשב רק בקשתות ירוקות וסגולות על צומת *n*, ובנה את הקבוצות *Ls* באופן טבעי.
7. עבור כל קשת $e \in E$,
8. אם צבעה שחור צבע אותה בירוק
9. אחרת, אם צבעה סגול הסר אותה מהגרף
10. הסר את צומת *n*
11. החזר את הקבוצות *Ls*.

נראה נכונות :

ניח בשלילה כי קיים *i* עבורו האלגוריתם מחזיר קבוצה $L_S(i)$ אשר מכילה צומת שמרחקה מקבוצה S אינו *i*.
האלגוריתם בונה גרף חדש, המורכב רק מקשתות ירוקות וסגולות, המצמצם את כל קבוצה *S* לצומת *n* מבלי לפגוע במרחקים, לפי לטענה 1. בשורה 6, ביצענו רדוקציה לאלגוריתם *BFS* רגיל, הפועל על הגרף החדש שבנינו.
מההנחה כי קיים *i* שמחזיר קבוצה $L_S(i)$ לא נכונה, נסיק כי אלגוריתם *BFS* שגוי, בסתירה לסעיף 3.3 עמ' 86.

ניתוח זמן ריצה :

שורה 1 רצה על כל הקשתות - $|E|$

שורות 3-5 רצות על כל הצמתים ב - *S*, ועל כל הקשתות מכל צומת כזו - $|E|$ + $|S|$

שורה 6 מבצעת *BFS* בעלות *|V-S|*+$|E|$

ובשורות 7-9 רצים על כל הקשתות $|E|$.

מכאן שסיבוכיות האלגוריתם הינו $O(|E| + |S| + |E| + |E| + |V - S| + |E|) = O(E + \max\{|S|, |V - S|\})$

כנדרש.

שאלה 3

נתון גרף לא מכוון $G = (V, E)$

טענה 1: בגרף לא מכוון קשיר, אם אין אף מעגל, אין שום דרך לכוון את הקשתות כך שלכל קודקוד תהייה דרגת כניסה גבוהה מאפס.

הוכחה: נניח כי קיימת דרך לכוון את הקשתות כך שנקבל לכל צומת דרגה גבוהה מאפס.
נכוון את הגרף בהתאם, ונקבל בחזרה גרף מכוון ללא מעגלים.
לפי 3.19 עמ' 110 קיימת צומת *v* אליה לא נכנסת אף קשת, כלומר דרגת הכנסה שלה היא אפס, בסתירה.

טענה 2: בגרף לא מכוון קשיר, מספיק מעגל אחד כדי שנוכל לכוון את הקשתות כך שלכל קודקוד דרגת כניסה גבוהה מאפס.

הוכחה: יהיו *v1,v2* צמתים סמוכים במעגל. נכוון את הקשת *{v1, v2}* להיות *(v1,v2)* (מ *v1* ל *v2*).
נבצע *BFS* מצומת *v2* לכל שאר העץ כאשר אנו מכוונים כל קשת מכיוון האב אל כיוון הבן, מלבד קשתות שכבר כוונו (לדוגמא קשת *(v1,v2)*). קיבלנו כי *v2* הוא שורש של עץ סריקת ה *BFS* של הגרף הקשיר, ולכן הוא מגיע לכל צומת בגרף, כולל צומת *v1*, שלא דרך קשת *(v1,v2)* בגלל המעגל. לכל צומת בעץ מלבד השורש יש אב, ולכן דרגת הכניסה של כל צומת מלבד האב גדולה מאפס, וכן יש את הקשת *(v1,v2)* אשר נכנסת אל האב, ומכאן שגם לאב יש דרגה גבוהה מ-0. וקיבלנו כי לכל הצמתים בגרף הקשיר יש דרגת כניסה גדולה מ-0.
אם נשארו עוד קשתות לא מכוונות, ניתן לכוון אותן כרצוננו, שכן אנו כבר עונים על התנאי.

בעזרת הטענות הנ"ל נבנה את האלגוריתם:

1. מצא את כל רכיבי הקשירות של *G*
2. לכל רכיב קשירות
3. אם מספר הקשתות קטן ממספר הצמתים פחות 1, אזי החזר "לא ניתן לכוון את הגרף", לפי טענה 1.
4. אחרת, כוון את הקשתות לפי האלגוריתם המתואר בהוכחת טענה 2.
5. החזר את כיווני הקשתות.

שאלה 4:

א. נתון G גרף בלתי מכוון:

נרשום אלגוריתם המחזיר מעגל אחד לפחות בגרף, אם קיים :
1. מצא את כל רכיבי הקשירות של G
2. לכל רכיב קשירות C
3. אם מספר הקשתות גדול או שווה למספר הצמתים אזי (מצאנו רכיב עם מעגל):
4. נבחר צומת $c \in C$, ונבצע ממנה BFS מותאם, אשר בפעם הראשונה שהוא נתקל בצומת בא הוא כבר ביקר, הוא מחזיר את הקשת המחברת בין הצומת הזו לצומת הקודמת $(v1,v2)$
5. נטפס במעלה עץ הסריקה החלקי שנוצר מ $v1$ עד שנגיע אל השורש, ונצבע כל צומת בדרך בצבע ירוק.
6. נטפס במעלה עץ הסריקה החלקי שנוצר מ $v2$ עד שנגיע אל צומת הצבועה בירוק, נצבע אותה בצבע כתום ונוסיף כל צומת כזו <u>לסוף</u> רשימה L.
7. נטפס שוב במעלה העץ מ $v1$ עד שנגיע לצומת כתומה, ונוסיף את הצמתים הללו <u>לתחילת</u> הרשימה L.
8. החזר את רשימה L
9. החזר "אינו מכיל מעגל"

נראה נכונות :

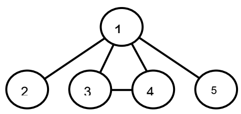
בשורה 3 אנו מוצאים רכיב קשירות בעל מעגל, לפי טענה 3.2.
כמו כן, סריקת ה BFS כפי שהגדרנו תפסק ברגע הראשון בו ימצא המעגל, ונקבל חזרה שני צמתים סמוכים המהווים חלק מהמעגל. ונטפס עד אשר נמצא את הצומת הראשונה שהיא אב קדמון של הצמתים הנ"ל, ראשית על ידי צביעה של הצמתים במסלול בירוק, ושנית על ידי חיפוש צומת ירוקה, והפיכתה לכתומה. כל שנותר הוא להוסיף את הצמתים הנ"ל לרשימת צמתים בסדר הנכון על מנת לקבל את המעגל שלנו, ואכן, אנו מתחילים מ $v2$ ומוסיפים את כל הצמתים לסוף הרשימה, מכאן שהאיבר הראשון ברשימה הוא $v2$ והאיבר האחרון הוא האיבר הכתום.
לאחר, מתחילים מ $v1$ ומוסיפים את כל הצמתים לתחילת הרשימה, והאיבר הראשון גם הוא האיבר הכתום. אם נקרא לאיבר הכתום o, הרי שהרשימה נראית כך: $o,\ldots,v1,v2,\ldots o$ ולכן סידרנו את הצמתים נכון ברשימה וקיבלנו מעגל.

זמן ריצה :

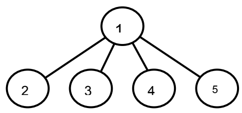
מציאת רכיבי הקשירות לוקחת $O(m+n)$ לפי סעיף 3.3 עמ' 102
יש לא יותר מ n רכיבי קשירות, על כל אחת מבצעים בדיקה בזמן קבוע ($m>=n$), ולכן לולאה זו רצה בזמן $O(n)$
ביצוע BFS רגיל לוקח $O(m+n)$ אולם, בפועל, עם השינוי שהכנסנו, הוא לוקח $O(n)$ (כי מפסיקים ברגע שחוזרים על צומת).
וטיפוס במעלה העץ לוקח $O(n)$, והוספה לרשימה דו מקושרת לוקחת $O(1)$.
לסיכום, האלגוריתם מתבצע בזמן $O(m+n)$

כנדרש.

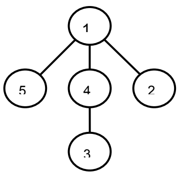
ב. דוגמא נגדית:

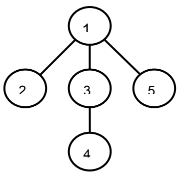


עץ פורש:



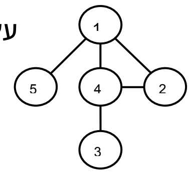
שני עצי DFS אפשריים שמתחילים ב 1:



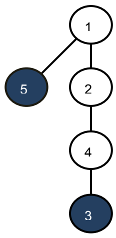


העץ הפורש אינו גרף תשתית של אף עץ DFS רלוונטי של הגרף.

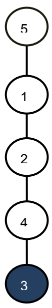
ג. דוגמא נגדית:



עץ DFS ראשון:



עץ DFS שני:



אין להם אותו מספר עלים

כנדרש!

שאלה 5:

נתון גרף לא מכוון קשיר וללא מעגלים $T = (V, E)$.

יהיו $v1, v2, x \in V$

טענה 1: בין שלושת המסלולים *(x,v1), (x,v2), (v1,v2)* קיימת צומת *c* על המסלול *(v1,v2)* המשותפת לשלושת המסלולים.
הוכחה: אם נניח שאין אף צומת על המסלול *(v1,v2)* המשותפת לשלושת המסלולים, נוכל ליצור מעגל על ידי המסלול הבא: *(x,v1) -> (v1,v2) -> (v2,x)* (הגרף אינו מכוון, ולכן קיום *(x,v2)* מבטיח קיום *(v2,x)*), וזו בסתירה לנתון כי ב *T* אין מעגלים.

יהיו $\mathrm{v1}, \mathrm{v2} \in \mathrm{V}$ קצוות קוטר בגרף *T*
טענה 2: לכל $x \in V$, הצומת הרחוקה ביותר מ *x* חייבת להיות *v1* או *v2*.
הוכחה: בלי הגבלת הכלליות, נניח $\mathrm{d}(\mathrm{x}, \mathrm{v1}) > d(x, v2)$,
ונניח כי יש צומת $q \in V$ שאינה על הקוטר, כך ש $\mathrm{d}(\mathrm{x}, \mathrm{q}) > d(x, v1)$.
מטענה 1, קיימת צומת *c* המשותפת למסלולים $(x, v1), (x, q), (v1, q)$ (שאינה בהכרח על הקוטר)
ולכן $d(x, q) = d(x, c) + d(c, q) > d(x, c) + d(c, v1) = d(x, v1)$
כלומר, $d(c, q) > d(c, v1)$
ונקבל $d(c, q) + d(c, v1) > d(c, v1) + d(c, v1) > d(v1, v2)$ $[*]$
כלומר $d(v1, q) > d(v1, v2)$
בסתירה לכך ש *(v1,v2)* הוא קוטר.

מטענה 2, נוכל לבנות אלגוריתם ליניארי למציאת הקוטר.

מצא קוטר (*G*)

1. בחר $x \in V$ כלשהו
2. מצא את הצומת הרחוקה ביותר מ-*x* ושים אותה ב *y* (*y* הוא קצה אחד של הקוטר, לפי טענה 2)
3. מצא את הצומת הרחוקה ביותר מ-*y* ושים אותה ב *z* (*z* הוא הקצה השני של הקוטר, לפי הגדרת הקוטר)
4. החזר $\mathrm{d}(\mathrm{y}, \mathrm{z})$

למציאת הצומת הרחוקה ביותר מצומת *x* כלשהי, נשתמש באלגוריתם *BFS* אשר מחזיר את העלה הנמוך ביותר בעץ הסריקה, וכאשר זה רץ על גרף לא מכוון קשיר וללא מעגלים, הרי שגרף זה הוא עץ בעצמו, ומכאן ש $|E| = |V| - 1$, והאלגוריתם מוצא אורך מכסימלי פעמיים לכל היותר, ומכאן סיבוכיות האלגוריתם היא *O(|V|)*.

כנדרש.

$[*]$ תחת ההנחה $d(x, v1) > d(x, v2)$ ועבור *c* הנתון, נוכל למצוא צומת *d* על מסלול *(c,v1), (c,v2), (v1, v2)* בעזרת טענה 1, ונטען כי $d(c, d) + d(d, v1) > d(c, d) + d(d, v2)$ כלומר $d(d, v1) > d(d, v2)$ ולכן $d(d, v1) + d(d, v1) > d(v1, d) + d(d, v2) = d(v1, v2)$, ובפרט $d(c, v1) + d(c, v1) > d(v1, v2)$.