1.11.2020 **_תכנון אלגוריתמים_**

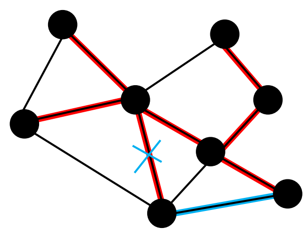
הרצאה 5: עץ פורש מינימום

Textbook: Cormen, Leiserson, Rivest, Stein. Introduction to Algorithms.

מרצה: אורי שטמר

<u>טענה:</u> יהי $T = (V, F)$ עץ פורש של גרף $G = (V, E)$, ותהי $e \notin F$. אזי הגרף $H = (V, F \cup \{e\})$ מכיל מעגל ולכל קשת e' במעגל זה מתקיים שהגרף $T' = (V, F \cup \{e\} \setminus \{e'\})$ הוא עץ פורש של G.

דוגמה:



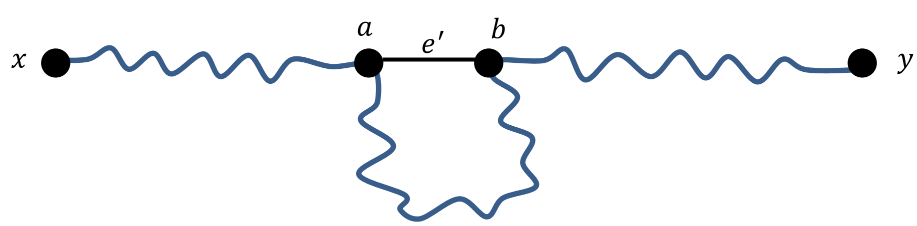
<u>הוכחת הטענה:</u>

H קשיר ומכיל $|V|$ קשתות
לכן H <u>איננו</u> עץ
לכן H מכיל מעגל (אחרת היה גרף קשיר חסר מעגלים שהוא עץ).

תהי e' קשת במעגל ב- H ונסמן $T' = (V, F \cup \{e\} \setminus \{e'\})$.

נראה ש- T' הוא עץ. ראשית, T' מכיל $|V| - 1$ קשתות, ולכן מספיק להראות שהוא קשיר. ואמנם:

יהיו x, y צמתים כלשהם בגרף. כיוון ש- T קשיר, אזי יש מסלול ב- T בין x ל- y. אם המסלול לא מכיל את e' אזי המסלול נמצא גם ב- T'. אחרת:



נסמן $e' = (a, b)$. הגרף H מכיל מעגל ש- e' היא צלע בו. לכן הגרף H מכיל מסלול מ- a ל- b אשר איננו כולל את הצלע e'. נסמנו $(a, u_1, \dots, u_\ell, b)$.

יהיה $P = (x, v_1, \dots, v_i, a, b, v_j, \dots, v_k, y)$ המסלול <u>הפשוט</u> בין x ל- y אשר נמצא ב- T. אזי המסלול
$$(x, v_1, \dots, v_i, a, u_1, \dots, u_\ell, b, v_j, \dots, v_k, y)$$
הוא מסלול מ- x ל- y הנמצא ב- T'. קיבלנו שלכל $x, y \in V$ יש מסלול מ- x ל- y ב- T', כלומר T' קשיר. כלומר T' קשיר ומכיל $|V| - 1$ קשתות ולכן הוא עץ.

מ.ש.ל.

שאלה: למה המסלול $(x, v_1, .., v_i, a, u_1, \dots, u_\ell, b, v_j, \dots, v_k, y)$ נמצא ב- T'?

תשובה: $(a, u_1, \dots, u_\ell, b)$ נמצא ב- T' כי הוא לא כולל את e' וזו הצלע היחידה שנמצאת ב- H ולא ב- T'. באופן דומה, המסלול P הוא פשוט ולכן מכיל את e' רק פעם אחת. את המופע שלה החלפנו במסלול ללא e' ולכן כעת כל המסלול לא מכיל את e'. כלומר זהו מסלול שנמצא ב- H ולא מכיל את e' ולכן נמצא ב- T'.

שאלה: איך אני יודע שהמסלול הזה הוא <u>פשוט</u>?

תשובה: זה לא מטריד אותנו. אנחנו כבר יודעים ש- T' מכיל $|V| - 1$ קשתות, ולכן כדי להראות שהוא עץ מספיק להראות שהוא קשיר (ולשם זה אנחנו לא חייבים לדבר על מסלולים פשוטים).

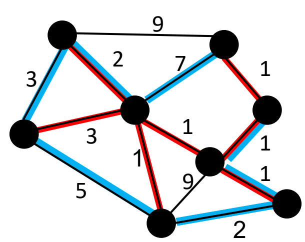
בעיית עץ פורש מינימום (עפ"מ)

<u>קלט:</u> גרף קשיר ולא מכוון $G = (V, E)$ ופונקציית משקל $w: E \to \mathbb{R}$

<u>פתרון חוקי:</u> עץ פורש של G

<u>יש למצוא:</u> עץ פורש שמשקלו מינימלי, כלומר עץ פורש $T = (V, F)$ כך ש- $\sum_{e \in F} w(e)$ מינימלי

<u>דוגמה:</u>
באדום: עפ"מ במשקל 10
בתכלת: עץ פורש אחר במשקל 21



מוטיבציה לבעיה: רוצים להקים רשתבין מחשבים ולשלם כמה שפחות על החיבורים

שימו לב: בפעם שעברה הוכחנו שלכל גרף קשיר ולא מכוון קיים עץ פורש, ולכן תמיד <u>קיים</u> פתרון לבעיה.

<u>רעיון כללי לאלגוריתם חמדן לבעיה:</u>

* אתחל $B \leftarrow \emptyset$ (קבוצת הצלעות שאנחנו בוחרים לפתרון שאנחנו בונים)
* כל עוד $|B| < |V| - 1$ בצע:
  - בחר קשר $e \in E \setminus B$ (כלומר קשת שעוד לא בחרנו) כך ש- $(V, B \cup \{e\})$ חסר מעגלים
  - $B \leftarrow B \cup \{e\}$

שאלה: איזה קשת נבחר בכל שלב?
תשובה: נבחר קשת עם משקל מנימלי (שלא סוגרת מעגל). זה מביא אותנו לאלגוריתם של קרוסקל:

האלגוריתם של קרוסקל

<u>אתחול:</u> $B \leftarrow \emptyset$, $C \leftarrow E$

<u>צעד:</u> כל עוד $|B| < |V| - 1$ בצע:

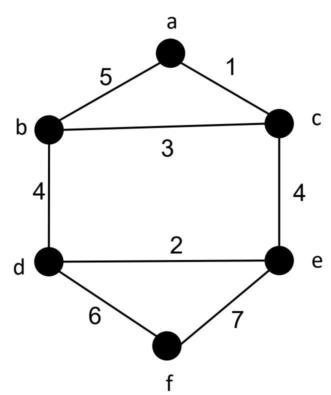
- בחר קשת $e \in C$ עם משקל מינימלי
- $C \leftarrow C \setminus \{e\}$
- אם $(V, B \cup \{e\})$ לא מכיל מעגל אזי $B \leftarrow B \cup \{e\}$

<u>החזר:</u> (V, B)

<u>אבחנה:</u> אם האלגוריתם עוצר אזי הוא מחזיר <u>עץ פורש</u> (כי הוא יהיה חסר מעגלים ויכיל $|V| - 1$ קשתות).

שאלה: למה האבחנה מנוסחת ככה "אם האלגוריתם עוצר..."?
תשובה: כי כרגע עדיין לא ברור לנו למה נקבל ש- $|B| = |V| - 1$ לפני שיגמרו הקשתות ב- C. אנחנו נראה שאם גרף הקלט הוא <u>קשיר</u> אז האלגוריתם תמיד עוצר ומחזיר עפ"מ.

<u>דוגמת ריצה:</u>



בכל שלב האלגוריתם בוחר את הקשת הקלה ביותר שלא סוגרת מעגל.
בהתחלה (a, c), אח"כ (d, e), אח"כ (b, c),
אח"כ אחת מ- $(b, d), (c, e)$. נניח למשל שהוא בחר את (b, d).
אח"כ את (d, f)

<u>הוכחת נכונות:</u>

<u>טענת עזר:</u> יהיה G גרף קשיר. בכל שלב באלגוריתם קיים עץ פורש מינימום שמכיל את הקשתות ב- B.

בפרט, לא יתכן מצב שבו $C = \emptyset$ אבל $|B| < |V| - 1$ כי אז ז"א שכל קשת מחוץ ל- B סוגרת מעגל ולכן לא ניתן להרחיב את B לעץ פורש. לכן זה בפרט יגיד שהאלגוריתם תמיד עוצר ומחזיר עץ פורש.

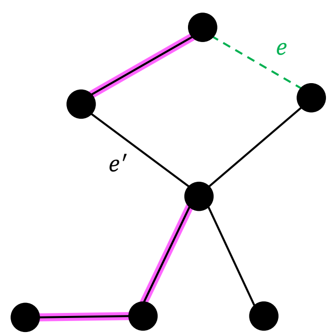
<u>הוכחת טענת העזר:</u> באינדוקציה על מספר הקשתות ב- B.

<u>בסיס:</u> $|B| = 0$ כלומר $B = \emptyset$ וכל עפ"מ מכיל את B

<u>צעד:</u> נניח שאחרי שהכנסנו $i - 1$ קשתות קיים עץ פורש מינימום $T = (V, F)$ שמכיל את B.
תהי e הקשת ה-i-ית שהאלגוריתם מוסיף ל- B.

<u>מקרה פשוט:</u> אם $e \in F$ אזי T עונה לדרישות, כלומר מכיל את $B \cup \{e\}$, כלומר את i הקשתות הראשונות שהאלגוריתם בחר.

<u>מקרה פחות פשוט:</u> $e \notin F$. נסתכל על $H = (V, F \cup \{e\})$.
H מכיל מעגל. הגרף $(V, B \cup \{e\})$ לא מכיל מעגל.
לכן קיימת קשת e' במעגל ב- H <u>שאינה</u> נמצאת ב- $(V, B \cup \{e\})$.



דוגמה:

בשחור: העפ"מ T המובטח מהנחת האינדוקציה

בורוד: הקשתות ב- B עד עכשיו

בירוק: הקשת הבאה שהאלגוריתם מוסיף

הקשת e' נמצאת על המעגל הנסגר כתוצאה מהוספת e, ולא נמצאת ב- B

הקשת e' שייכת ל- T שמכיל את B. לכן $B \cup \{e'\}$ לא מכיל מעגל.
האלגוריתם של קרוסקל בחר קשת עם משקל מינימלי שלא סוגרת מעגל, ולכן $w(e) \leq w(e')$.
נסתכל על $T' = (V, F \cup \{e\} \setminus \{e'\})$. לפי הטענה שהוכחנו בתחילת השיעור, זהו עץ פורש. משקלו הוא

$$w(T') = w(T) + w(e) - w(e') \leq w(T)$$

כלומר T' עץ פורש מינימום.
בנוסף, T' מכיל את כל הקשתות ב- $B \cup \{e\}$ (כי $e' \notin B$).

מ.ש.ל.

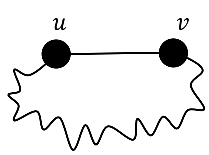
משפט: האלגוריתם של קרוסקל מחזיר עץ פורש מינימום בגרף.

הוכחה: אם האלגוריתם הגיע לשלב הסיום, אזי הוא מחזיר גרף חסר מעגלים שמכיל $|V| - 1$ קשתות. לכן הוא מחזיר עץ פורש שלפי טענת העזר מוכל בעץ פורש מינימום ולכן הוא עץ פורש מינימום.

נראה שהאלגוריתם מסתיים, כלומר בוחר $|V| - 1$ קשתות:

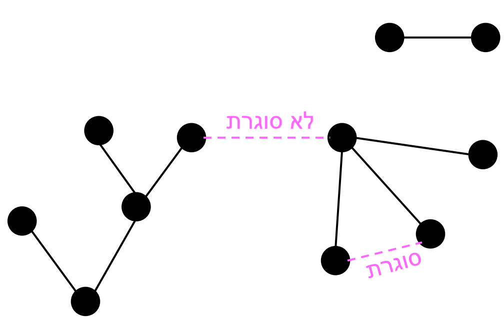
נניח כי בשלב מסויים $|B| < |V| - 1$ ונראה כי אז $|C| > 0$.
לפי טענת העזר קיים עץ פורש T המכיל את קשתות B.
מכיוון שב- T יש $|V| - 1$ קשתות, קיימת קשת e המופיעה ב- T אבל לא ב- B.
אזי $e \in C$ ולכן $C \neq \emptyset$.
(אחרת ז"א שכבר בחנו את הצלע e והחלטנו לא להכניס אותה ל- B כי היא סגרה מעגל, אבל זה לא יתכן כי T מכיל את הקשתות ב $B \cup \{e\}$ וב- T אין מעגלים)

ניתוח זמן ריצה:



איך בודקים אם קשת סוגרת מעגל?
(u, v) סוגרת מעגל אם בגרף (V, B) יש מסלול בין u ל- v.
הבדיקה האם יש מסלול היא ע"י BFS.
הגרף (V, B) מכיל פחות מ- $|V|$ קשתות ולכן הרצת BFS אחת תעלה $O(|V|)$.
סה"כ נריץ $O(|E|)$ פעמים BFS ונקבל אלגוריתם בסיבוכיות $O(|E| \cdot |V|)$.

נראה מימוש יעיל יותר:



בשלב מסויים של האלגוריתם:

קשת בין 2 תתי עצים לא סוגרת מעגל.
קשת בתוך אותו תת העץ כן סוגרת מעגל.

בכל שלב נחזיק את הקבוצות של תתי העצים, וכדי לבדוק האם

קשת סוגרת מעגל נבדוק האם 2 הקצוות שלה בתוך אותה קבוצה.
- בתחילת הריצה: כל צומת יהיה בקבוצה בגודל 1.
- כאשר מוסיפים קשת (u,v) מבצעים $\text{Union}(u,v)$
- בדיקה אם קשת סוגרת מעגל: בדיקה האם הקבוצה של u היא הקבוצה של v.

<u>מימוש מדוייק:</u>
1. מיין את הקשתות עפ"י משקלם בסדר לא יורד.
2. לכל $v \in V$ בצע $\text{MakeSet}(v)$
3. $B \leftarrow \emptyset$
4. עבור על כל הקשתות ב- E עפ"י סדר המיון:
   - תהי (u,v) הקש הבאה במיון
   - אם $\text{FindSet}(u) \neq \text{FindSet}(v)$ אזי:
     $B \leftarrow B \cup \{(u,v)\}$
     $\text{Union}(u,v)$
5. החזר (V,B)

<u>נשתמש במבנה הנתונים UnionFind:</u>
* פעולת MakeSet מתבצעת בזמן $O(1)$.
* פעולת FindSet ופעולת Union מתבצעות בזמן $O(\log^* k)$ כאשר k הוא מספר האיברים הכולל ($k = |V|$ במקרה שלנו).

ישנם $|V|$ פעולות MakeSet.
לכל קשת נבצע לכל היותר 2 פעולות FindSet, כלומר סה"כ $O(|E|)$ פעולות FindSet.
מספר פעולות ה- Union הוא $O(|V|)$.
כלומר סה"כ זמן $O(|E| \cdot \log^*|V|)$ לאחר האתחול.
האתחול (מיון) מתבצע בזמן $O(|E| \cdot \log|E|)$ ולכן סה"כ זמן ריצה $O(|E| \cdot \log|E|)$.

שאלה: מזה $\log^* n$?
תשובה: מספר הפעמים שצריך להוציא $\log(\cdot)$ עד שמקבלים מספר קטן שווה 1.

האלגוריתם של פרים

האלגוריתם של קרוסקל כל פעם בוחר את הקשת הקלה ביותר (שלא סוגרת מעגל) ולאורך הריצה קבוצת הקשתות שהוא בחר עד כה "נראות בבלאגן". באלגוריתם הבא שנראה, בכל שלב נתחזק עץ על קבוצת הקודקודים שבחרנו עד כה. הכלל החמדן שלנו יהיה להסתכל על כל הקשתות שיוצאות מתוך קבוצת הקודקודים שבחרנו עד כה ומתוכם ניקח את הקשת הקלה ביותר.

<u>אתחול:</u> $S \leftarrow \{r\}$ עבור צומת שרירותי r (S היא קבוצת הקודקודים שבחרנו עד עכשיו)
$B \leftarrow \emptyset$ (B היא קבוצת הצלעות שבחרנו עד עכשיו)

<u>צעד:</u> כל עוד $|B| < |V| - 1$:
- מצא קשת מינימלית במשקלה $e = (v,w)$ כך ש- $v \in S$ ו- $w \notin S$.
- $B \leftarrow B \cup \{e\}$
- $S \leftarrow S \cup \{w\}$

<u>החזר:</u> (V,B)

<u>דוגמת ריצה:</u>