*תכנון אלגוריתמים* 4.11.2020

# הרצאה 6: עץ פורש מינימום

מרצה: אורי שטמר

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

## האלגוריתם של פרים – המשך

**<u>הוכחת נכונות</u>**

**<u>טענה נשמרת:</u>** בכל שלב באלגוריתם קיים עץ פורש מינימום שמכיל את כל בחירות האלגוריתם.

**<u>הוכחה:</u>** באינדוקציה של מספר הקשתות ב- $B$.

<u>בסיס:</u> טריוויאלי

<u>צעד:</u> נניח כי קיים עץ $T = (V, F)$ שמכיל את $i - 1$ הקשתות הראשונות שהאלגוריתם בחר, ותהי $e = (v, w)$ הקשת ה- $i$ שהאלגוריתם בחר.

<u>מקרה פשוט:</u> $e \in F$. לא צריך לעשות שום דבר.

<u>מקרה מעניין:</u> $e = (v, w) \notin F$

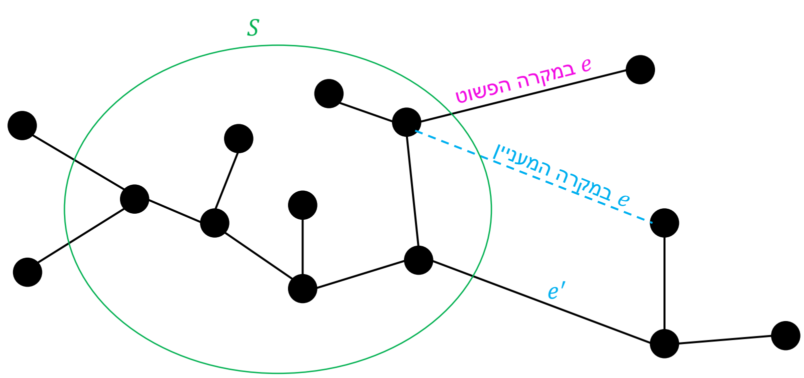
בעץ $T$ יש מסלול מ- $v$ ל- $w$. נסמנו:

$$v = v_1, v_2, v_3, \dots, v_{\ell-1}, v_\ell = w$$

נשים לב ש- $v_1 = v \in S$ ו- $v_\ell = w \notin S$.

יהי $j$ אינדקס כך ש- $v_j \in S$ ו- $v_{j+1} \notin S$.
כלומר הקשת $e' = (v_j, v_{j+1})$ מחברת בין צומת ב- $S$ לבין צומת שלא ב- $S$.

דוגמה:
בשחור – העץ $T = (V, F)$



האלגוריתם בחר בקשת $e = (v, w)$ ולא בקשת $e' = (v_j, v_{j+1})$, ולכן $w(v, w) \leq w(v_j, v_{j+1})$.

נסתכל על $T' = (V, F \cup \{e\} \setminus \{e'\})$

עפ"י הטענה מהשיעור הקודם, $T'$ הוא עץ פורש. בנוסף,

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

ולכן $T'$ הוא עפ"מ המכיל את $B \cup \{e\}$ כנדרש.

מ.ש.ל.

**<u>משפט:</u>** אם $G$ קשיר אזי האלגוריתם של פרים מחזיר עפ"מ.

**<u>הוכחה:</u>** בכל שלב קיימת לפחות קשת אחת שיוצאת מ- $S$ (כל עוד $|S| < |V|$, כי הגרף קשיר). לכן האלגוריתם לא נתקע.

עפ"י הטענה הנשמרת, בסוף הריצה קיים עפ"מ שמכיל את הבחירות שהאלגוריתם עשה. לכן הגרף המוחזר $(V, B)$ חסר מעגלים. בסיום הריצה $|B| = |V| - 1$ ולכן הגרף המוחזר $(V, B)$ הוא עץ (חסר מעגלים ומכיל $|V| - 1$ קשתות).

כלומר, הגרף המוחזר $(V, B)$ הוא עץ פורש וקיים <u>עפ"מ</u> שמכיל אותו ולכן חייב להיות שווה לו (אם עץ פורש מוכל בעץ פורש אז הם שווים).

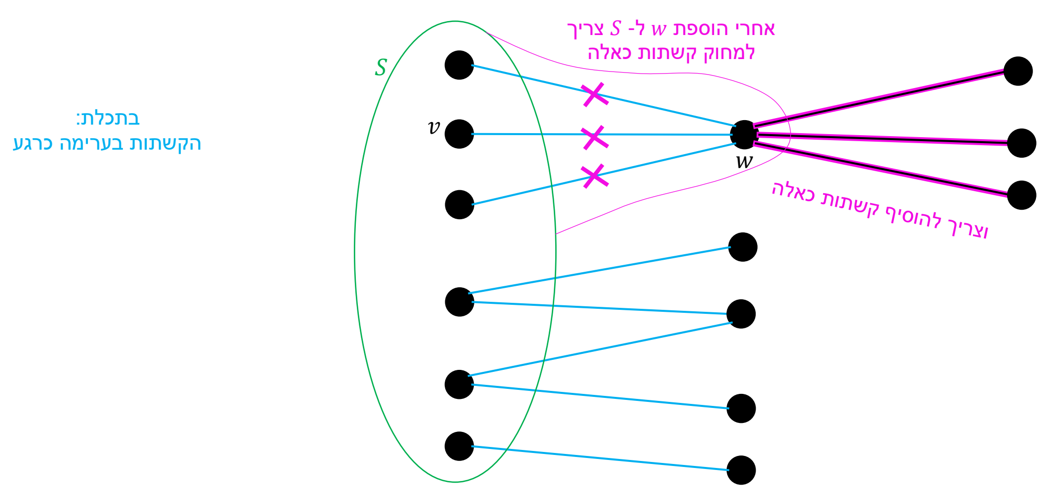
**<u>איך נממש את האלגוריתם?</u>**

**<u>מימוש נאיבי</u>**
בכל שלב נעבור על כל הקשתות ונבדוק לגבי כל קשת אם צד אחד שלה ב- $S$ וצד שני לא. ניקח קשת עם משקל מינימלי מבין הקשתות שיוצאות מ- $S$.
מחיר כל צעד: $O(|E|)$
מספר הצעדים, כלומר מספר הפעמים שנוסיף קשת: $O(|V|)$
סה"כ: $O(|V| \cdot |E|)$

במימוש הנאיבי הזה בכל שלב עברנו על כל הקשתות וחיפשנו קשת שיוצאת מ- $S$ עם משקל מינימלי. איזה מבנה נתונים יכול לעזור לנו כאן למצוא בכל שלב קשת כזאת ביעילות? ערימה! נרצה לדאוג שבכל שלב הערימה תכיל בדיוק את כל הקשתות שיוצאות מ- $S$.

**<u>רעיון למימוש יעיל יותר</u>**
נחזיק את כל הקשתות בין $S$ ל- $V \setminus S$ בערימה. לאחר שנבחר קשת מינימלית $(v, w)$ נוציא אותה מהערימה, נוסיף את כל הקשתות שיוצאות מ- $w$ אל $V \setminus S$, ונוציא מהערימה את כל הקשתות שיוצאות מ- $w$ אל $S$.

בציור:



**<u>מימוש מדוייק:</u>**
נחזיק את כל הקשתות בין $S$ ל- $V \setminus S$ בערימה (המפתח הוא המשקל שלהן).

<u>אתחול:</u>
- $S \leftarrow \{r\}$
- $B \leftarrow \emptyset$
- לכל שכן $w$ של $r$ נכניס את $(r,w)$ לערימה

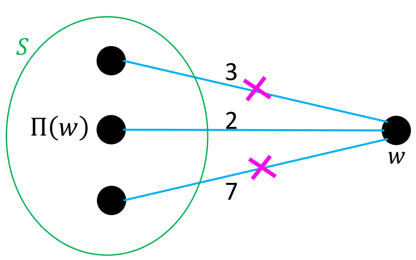
<u>צעד:</u> כל עוד $|B| < |V| - 1$:
- הוצא מהערימה קשת $e$ עם משקל מינימום
- נסמן $e = (v,w)$ כך ש- $v \in S$ ו- $w \notin S$
- לכל שכן $x$ של $w$ בצע:
  - אם $x \in S$ אזי הורד את $(x,w)$ מהערימה
  - אם $x \notin S$ אזי הוסף את $(x,w)$ לערימה
- $B \leftarrow B \cup \{e\}$
- $S \leftarrow S \cup \{w\}$

<u>החזר:</u> $(V,B)$

**<u>זמן ריצה:</u>**
בערימה יש לכל היותר $|E|$ איברים. לכן, כ"א מהפעולות הוצאת איבר מינימלי, מחיקה, הכנסה מתבצעת בזמן $O(\log|E|)$.
- מספר פעולות ExtractMin הוא $O(|V|)$.
- כל קשת תכנס לכל היותר פעם אחת לערימה (כאשר הצד הראשון שלה נכנס ל- $S$) ותצא פעם אחת מהערימה (כאשר הצד השני שלה נכנס ל- $S$)
- סה"כ מספר פעולות ה- Insert, Delete הוא $O(|E|)$

סה"כ סיבוכיות $O(|E| \cdot \log|E|)$

**<u>מימוש עוד יותר יעיל:</u>**




נסתכל על צומת $w$ מחוץ ל- $S$. יתכן שיש לו הרבה שכנים ב- $S$.
במימוש הקודם שמרנו בערימה את כל הקשתות בין $w$ לצמתים ב- $S$.
אבל בעצם לא צריך לשמור את כולם. מספיק לשמור רק את הקלה...

**הפעם הערימה שלנו תכיל צמתים (במקום קשתות) והמפתח בערימה של צומת $S \not\ni w$ יהיה משקל הקשת המינימלית שמחברת את לצומת ב- $S$ (נקרא לזה "המרחק" של $w$ מ- $S$)**

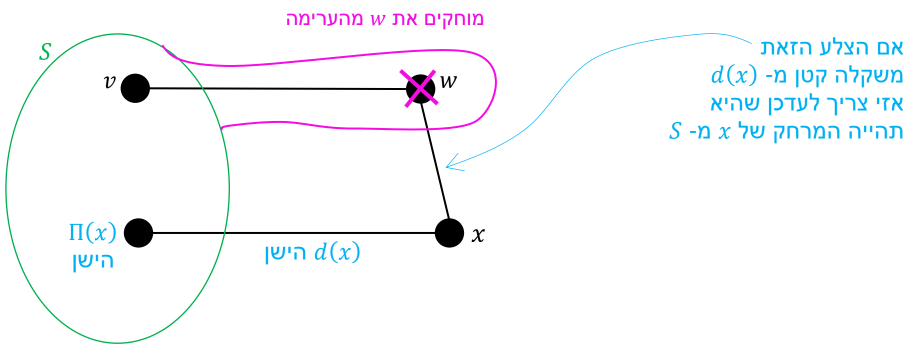
**<u>הגדרה:</u>** לכל צומת $w \notin S$ נגדיר את <u>המרחק שלו</u> מ- $S$ להיות
$$d(w) = \min\{w(v,w) :\ v \in S,\ \ (v,w) \in E\}$$
כאשר אם אין קשת כנ"ל אזי $d(w) = \infty$.

**<u>סימון נוסף:</u>** עבור צומת $w \notin S$ כך ש- $d(w) \neq \infty$ נסמן $\Pi(w) = v$ עבור קודקוד $v$ כך ש-
$$w(v,w) = d(w), \qquad (v,w) \in E, \qquad v \in S$$

**איך נעשה את האתחול?**

נגדיר $d(r)=0$ ו- $d(\text{כל השאר})=\infty$

**איך נעשה את העדכון?**




אחרי הכנסת $w$ ל- $S$ צריך לעדכן את $d(x)$ לכל $x\notin S$ שכן של $w$, על ידי מינימום בין $d(x)$ הקודם לבין המשקל של $(w,x)$.

**המימוש בצורה מדוייקת:**

אתחול:

- לכל $v\in V$ בצע $d(v)=\infty$, $\Pi(v)=\text{NIL}$
- $d(r)=0$
- $S\leftarrow\emptyset$
- $B\leftarrow\emptyset$
- נבנה ערימה המכילה את כל הצמתים ב- $V$ והמפתח שלהם הוא המרחק ל- $S$, כלומר $d(\cdot)$

צעד: כל עוד $|B|<|V|-1$:

- $w=\text{ExtractMin}$
- $S\leftarrow S\cup\{w\}$
- אם $w\neq r$ אזי $B\leftarrow B\cup\{(\Pi(w),w)\}$
- לכל שכן $x$ של $w$ בצע:
  - אם $x\notin S$ וגם $w(w,x)<d(x)$ אזי
    - $d(x)\leftarrow w(w,x)$ (שימו לב, צעד זה כולל עדכון מפתח בערימה)
    - $\Pi(x)\leftarrow w$

החזר: $(V,B)$

**ניתוח סיבוכיות:**

- מספר פעולות ה ExtractMin הוא $O(|V|)$ כי כל צומת תצא פעם אחת מהערימה. עלות כל פעולה בודדת כזאת היא $O(\log|V|)$
- ישנם $O(|E|)$ פעולות עדכון מפתח. עלות כל פעולה בודדת $O(\log|V|)$

סה"כ: $O(|E|\cdot\log|V|)=O(|E|\cdot\log|E|)$

זכרו כי תמיד מתקיים ש- $|E|\le|V|^2$ ולכן $\log|E|\le 2\log|V|$. בנוסף, במקרה שלנו הגרף קשיר ולכן $|E|\ge|V|-1$

כלומר, לפחות בנתיים נראה שלא הרווחנו ביעילות לעומת המימוש היעיל הקודם שעשינו (כאשר הערימה הכילה צלעות)...

**<u>אבחנה:</u>** $d(v)$ יכול רק <u>לקטון</u> במהלך הריצה.

קיים מבנה נתונים הנקרה <u>ערימת פיבונאצ'י</u> שבו פעולת ExtractMin מתבצעת בזמן $O(\log|V|)$, כמו בערימה רגילה, אבל פעולת DecreaseKey מתבצעת בזמן $O(1)$ בחשבון פחת. לכן $O(|E|)$ פעולות כאלה יעלו $O(|E|)$.

סה"כ סיבוכיות $O(|E| + V \cdot \log|V|)$