הרצאה 5

אלגוריתם Primהמשך:

תזכורת - האלגוריתם:

$1.T \leftarrow \emptyset,\ S \leftarrow \{s\}$
$2.$**while**$S \neq V$:
    $2.1\ e = (u,v)$ **e is the lightest edge crossing** $(S, \overline{S})$ **assuming** $u \in S, v \notin S$
    $2.2\ T \leftarrow T \cup \{e\}, \quad S \leftarrow S \cup \{v\}$
$3.$**Return** T

כיצד נממש את האלגוריתם Prim?

נשמור את צמתי S בערימת מינימום. הערך והמפתח של צומת v בערימה יהיה משקל הקשת הקלה ביותר שנוגעת בו ומחברת אותו ל-S (אם לא קיימת קשת כזו ערך המפתח יהיה ∞).

אתחול:

המפתח של S יהיה O.

המפתח של כל צומת אחר יהיה ∞.

צעד:

ברגע שמוציאים את צומת u מהערימה, מעדכנים את המפתחות של השכנים שלו שעדיין בערימה.

סיבוכיות:

$O(|E\log|V|)$

האלגוריתם של Kruskal:

1. מיין את הקשתות: $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m)$
   $T \leftarrow \emptyset$
2. מ-$i = 1$ עד m:
   אם $T \cup \{e_i\}$ לא מכיל מעגלים אזי $T \leftarrow T \cup \{e_i\}$.
3. החזר את T.

משפט:

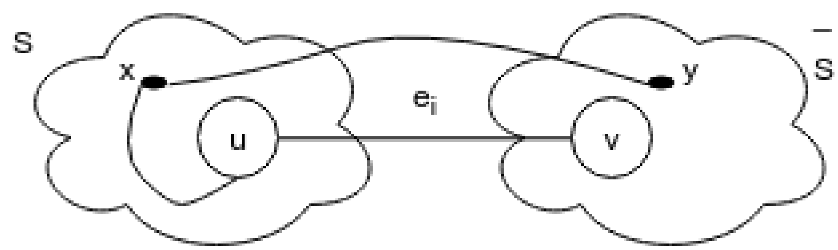
האלגוריתם של Kruskal מחזיר עץ פורש מינימום.

הוכחה:

נראה שהאלגוריתם של Kruskal הוא מימוש מסוים של השיטה הכללית. כלומר אם נצבע בכחול כל קשת שהאלגוריתם מוסיף ל-T ונצבע באדום כל קשת שהאלגוריתם לא מוסיף ל-T, נקבל הפעלה חוקית של הכללים.
ממחלק לשני מקרים:

1. ברגע שהאלגוריתם בוחן את $T \cup \{e_i\}$ מתקיים: $T \cup \{e_i\}$ מכיל מעגל C. נשים לב שבמעגל C כל הקשתות שייכות ל-T (פרט ל-e_i), כלומר כולן צבועות בכחול.
לפי הכלל האדום, ניתן לצבוע באדום את הקשת הכבדה ביותר ב-C מבין הקשתות הלא-צבועות. e_i היא הקשת היחידה ב-C שאינה צבועה ולכן צביעתה באדום היא הפעלה חוקים של הכלל האדום.

2. ברגע שהאלגוריתם בוחן את הקשת e_i מתקיים: ב-$T \cup \{e_i\}$ לא נסגרת מעגל



נתבונן בתחום: $S = \{x | \text{there is a path from } x \text{ to } u \text{ in } T\}$

(א) $u \in S$ (מההגדרה).

(ב) $v \notin S$ נראה זאת: נניח השלילה כי $v \in S$
$\Leftarrow$ יש ב-T מסלול (כחול) בין u ו-v (הגדרת S)
$\Leftarrow$ $T \cup \{e_i\}$ מכיל מעגל וזו סתירה.

(ג) למה אין קשת כחולה שחוצה את S?
נניח בשילה שיש קשת כחולה (x, y) כך שמתקיים $x \in S, y \notin S$
$x \in S \Leftarrow$ יש מסלול כחול בין u ו-x. נשרשר למסלול זה את הקשת (x, y) וקבלנו מסלול כחול בין u ו-y וזו סתירה לכך ש-$y \notin S$.
<u>שאלה:</u> מדוע אין אף קשת e שחוצה את S, אינה צבועה וגם $w(e) < w(e_i)$?
אם זה קורה, eהייתה צריכה להיצבע באיטרציה קודמת וזו סתירה.
$\Leftarrow$ זו הפעלה חוקית של הכלל הכחול. ■

כיצד ניתן לממש את האלגוריתם ?

הרעיון: הכל איטרציה נרצה לשמור את רכיבי הקשירות של (V, T).
נשתמש ב-union find.
הפעולות שצריך בכל איטרציה:

1. לבדוק האם שני רכיבי קשירות באותה קבוצה.
2. אולי לאחד שני רכיבי קשירות.

$\Leftarrow$ סה"כ סיבוכיות: $O(|E| \log |V|)$.

מסלולים קלים ביותר

נתון:

גרף מכוון $G=(V,E)$, פונקציית משקל $W:E\to\mathbb{R}$

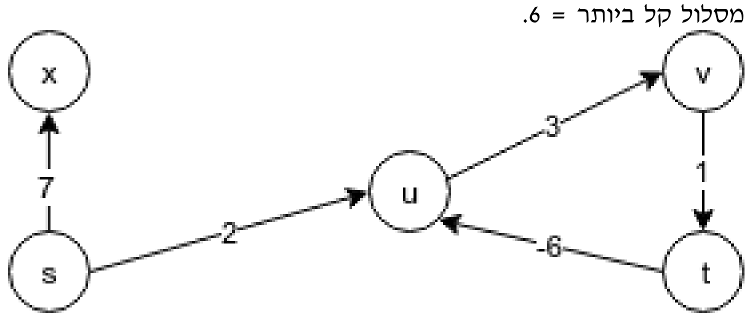
משקל של מסלול p מוגדר להיות סך משקלי הקשתות ב-p: $[w(p)\overset{\triangle}{=}\sum\limits_{e\in p}w(e)]$

מטרה:

בהינתן שני צמתים s ו-t, מהו המסלול הקל ביותר מ-s ל-t?

דוגמה:

מסלול קל ביותר = 6.



- ⋆ אם יש מעגל שלילי בגרף (מעגל שסכום משקלי הקשתות בו קטן ממש מאפס, אזי מרחקים קלים ביותר לא בהכרח מוגדרים).
- ⋆ הערה: המקרה ש- $w(e)=1\quad\forall e\in E$ נפתר ע"י BFS.

הבחנה:

אם p מסלול קל ביותר מ-u ל-v, כל תת-מסלול של p גם הוא קל ביותר.

הוכחה:

נסתכל על p:

$$p:\ u=u_0\overset{e_1}{\to}u_1\overset{e_2}{\to}u_2\to\cdots\to u_{k-1}\to u_k=v$$

נתבונן בתת-המסלול מ-u_i ל-u_j $(i<j)$.

מתקיים: $u\overset{p'}{\to}u_i\overset{p_{ij}}{\to}u_j\overset{p''}{\to}v$

$$w(p)=w(p')+w(p_{ij})+w(p'')$$

נניח בשלילה ש-p_{ij} אינו קל ביותר.

$\Leftarrow$ יש מסלול q מ-u_i ל-u_j כך ש-$w(q)<w(p_{ij})$

נבנה מסלול חדש מ-u ל-v באופן הבא: $u\overset{p'}{\to}u_i\overset{q}{\to}u_j\overset{p''}{\to}v$

שאורכו: $w(p)=w(p')+w(q)+w(p'')$

וזו סתירה ■

נסמן ב- $\delta(u,v)$ את אורך המסלול הקל ביותר מ-u ל-v:

$$\delta(u,v)=\begin{cases}\infty & v\text{ is not reachable from }u\\ -\infty & \text{"negative" circle reachable from }u\\ & \text{and v reachable from the circle}\\ \min\{w(p):p\text{=path from }u\text{ to }v\} & \text{otherwise}\end{cases}$$