17/12

29

בס״ד

נושא 7 - MST (Minimum Spanning Tree)

א) <u>הגדרות:</u>

• <u>גרף ממושקל</u> - זהו גרף שלכל הצלעות בו יש משקל, שהוא עלות המעבר בהן. המשקל יכול לייצג מרחק, זמן מעבר, אגרה על המעבר בקשת, וכו'. משקל קשת e נסמן $w(e)$.

• <u>עץ פורש</u> - זהו עץ פורש מתוך הגרף שמכיל את כל הצמתים בגרף. עץ פורש הוא כמובן חסר מעגלים (לפי הגדרת עץ), ומכיל n-1 קשתות.

• <u>עץ פורש מינימלי (MST)</u> - זהו עץ פורש שסך משקל כל הקשתות בו הוא מינימלי. $w(T)=\sum_{e\in T} w(e)$

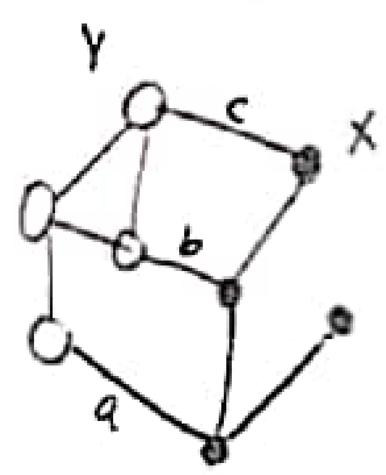
מהי האסטרטגיה בה נבנה עץ פורש מינימלי?

(1) נתחזק קבוצה של קשתות A כך ש-(V,A) הוא יער פורש של G. יער פורש הוא קבוצה של עצים המכילה את כל הצמתים. גם קבוצת כל הצמתים ללא קשתות כלל נקראת יער פורש, כי כל צומת הוא בעצמו עץ. בנוסף קיים עץ פורש מינימלי $T=(V,F)$ כך ש- $A\subseteq F$.

(2) כל עוד A הוא יער ולא עץ, נמצא "קשת בטוחה" שאם נוסיף אותה ל-A אזי A עדיין יהיה חלק מ-MST. כיצד למצוא קשת בטוחה נדון בסעיפים הבאים.

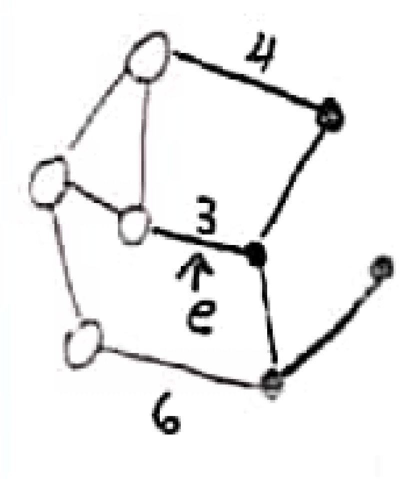
(3) נחזיר את העץ הפורש T.

ב) <u>חתך</u>: חתך (X,Y) של גרף $G=(V,E)$ הוא חלוקה של קבוצת הצמתים V לשתי קבוצות: X ו- $Y=V\setminus X$. כלומר אין צומת ב-V שהוא גם ב-X וגם ב-Y או שלא מופיע בשניהם. נאמר כי קשת (v,w) "חוצה את החתך" אם $v\in X$ ו- $w\in Y$. וכן ההיפך. נאמר כי החתך (X,Y) "מכבד" קבוצה של קשתות A אם אין קשת ב-A שחוצה את החתך (X,Y).



a,b חוצות את החתך. כל שאר הקשתות מכובדות ע"י החתך.

<u>משפט החתך</u>: יהי גרף $G=(V,E)$, ויהי קבוצה של קשתות A כך ש-A הוא תת קבוצה של איזשהו עץ פורש מינימלי של G. יהי (X,Y) חתך של G שמכבד את A, כלומר אין קשת ב-A שחוצה את (X,Y). ויהי e הקשת המינימלית מבין כל הקשתות שחוצות את (X,Y). אזי $A\cup\{e\}$ גם הוא תת הקבוצה של איזשהו עץ פורש מינימלי של G.



במילים אחרות כדי למצוא MST נצטרך בכל איטרציה למצוא חתך שמכבד את A, ואז למצוא את הקשת המינימלית שחוצה את החתך ולהוסיף ל-A. הוכחה בעמוד הבא.

17/12

30

בס"ד

<u>הוכחה:</u> נגדיר T עץ פורש מינימלי על G המכיל את A. נגדיר בנוסף קשת F שאינה נמצאת ב-A. עבור החתך (X,Y) המכבד את A, גם e וגם F חוצות את החתך, אך לפי הגדרת e מתקיים $w(e)\le w(F)$. נניח בשלילה כי $F\in T$ וכי $e\notin T$. האלגוריתם שלנו יבחר את e ולא את F, אזי נבנה עץ נוסף $T'=T-\{F\}\cup\{e\}$, שהוא עץ פורש עם e ללא F. מכיוון שמתקיים $w(T')\le w(T)$ ו-T הוא בהכרח מינימלי אזי גם T' חייב להיות עץ פורש מינימלי. אחרת נגיע לסתירה!

ג) <u>אלגוריתם קרוסקל (Kruskal)</u>

אלגוריתם קרוסקל כהו אחד משני אלגוריתמים המוצאים MST לפי העקרונות שלמדנו.

<u>פעולת האלגוריתם:</u>

(1) נאתחל קבוצה ריקה של קשתות $A=\emptyset$. — $O(1)$

(2) נאתחל מבנה נתונים Union-Find בשם S שמחזיק כל צומת בגרף $G=(V,E)$ בתור קבוצה זרה. — $O(n)$

(3) נמיין את כל הקשתות ב-E לפי המשקל שלהם מהקטן לגדול. — $O(m\log m)$

(4) נעבור על כל קבוצת הקשתות הממוינת. בכל קשת $e=(v,w)$, אם v ו-w בשני קבוצות זרות ב-S, כלומר $Find(v)\neq Find(w)$ אזי נוסיף את e ל-A ונאחד את v עם w. — $O((n+m)\alpha(n))$

```
If ( Find(v) ≠ Find(w) {
        A.add(e)
        Union(v,w)
}
```

(5) לאחר המעבר על כל הקשתות נחזיר את A. — $O(1)$

<u>הוכחת האלגוריתם:</u> זהו בעצם המימוש של משפט החתך. מוצאים את הקשת עם המשקל המינימלי בין שתי קבוצות זרות, המייצגים חתך כלשהו המכבד את A, ומוסיפים אותה ל-A.

<u>סיבוכיות האלגוריתם:</u> נסמן n מספר הצמתים ו-m מספר הקשתות. אתחול קבוצה ריקה מתבצע ב-$O(1)$. אתחול Union-Find על כל הצמתים ב-$O(n)$. מיון כל הקשתות ב-$O(m\log m)$. בשלב הרביעי, כל פעולות חיפוש ואיחוד על מבנה נתונים Union-Find הוא בזמן ריצה של $\alpha(n)$ שהיא הפונקציה ההפוכה לפונקציית אקרמן שגדלה מאוד מאוד לאט (אקרמן גדלה מאוד מהר). בתוך שלב זה נבצע m פעולות חיפוש, ו-n-1 איחודים, כמספר הקשתות בעץ פורש. לכן סה"כ נקבל $O((n+m)\alpha(n))$. מכיוון שצריך ש-$n-1\le m$, כי אחרת אין עץ פורש, נקבל $O(m\cdot\alpha(n))$, שהוא קטן מ-$O(m\log m)$.

מסקנה - סיבוכיות אלגוריתם קרוסקל היא <u>$O(m\log m)$</u>.

17/12 בס"ד

(3-1)

3) אלגוריתם פרים (prim)

פעולת האלגוריתם:

```
prim(G) {
  Q = V
  for each u∈Q
    key[u] = ∞
  key[s] = 0,
  while Q not empty {
    u = removeMin(Q)
    for each v∈adj[u] {
      if(v∈Q AND w(u,v) < key[v])
        key[v] = w(u,v)
      if(w(u,v) < w(e))
        A.add(e)
    }
  }
  return A
}
```

(1) נבנה קבוצה Q המכילה את כל הצמתים ב-G. $O(n)$

(2) נאתחל את ה-key של כל הצמתים ל-∞. $O(n)$

(3) נבחר צומת כלשהי s ונאתחל את ה-key ל-0, כדי לוודא שצומת זו תיבחר ראשונה בפועל. $O(1)$

(4) כל עוד Q לא קבוצה ריקה, נבחר את הצומת u עם ה-key המינימלי. נעבור על כל שכן של u. אם השכן v נמצא ב-Q וגם משקל הקשת אליו קטן מהמשקל של v, נעדכן את המשקל של v להיות משקל הקשת אליו. $O(m\log n)$

(5) בנוסף במעבר על כל השכנים נחפש את הקשת המינימלית, ונכניס אותה ל-A.

(6) נחזיר את קבוצת הקשתות A. $O(1)$

סיכום: האלגוריתם מתחיל מצומת שנבחרת באופן שרירותי ובכל איטרציה מוסיפה לעץ את הקשת בעלת המשקל המינימלי מבין כל הקשתות שיוצאות מצמתי העץ ולא סוגרות מעגל.

הוכחת האלגוריתם: בכל איטרציה יש לנו עץ, המכיל חלק מהקודקודים בגרף.. ניתן להסתכל על קבוצת הצמתים בעץ ושאר הצמתים שלא בעץ כעל חתך של G. בכל איטרציה אנו בוחרים את הקשת עם המשקל המינימלי היוצאת מהעץ שלא סוגרת מעגל. זהו בעצם מימוש של משפט החתך.

סיבוכיות האלגוריתם: כל הפעולות מלבד 4-5 הם בסיבוכיות של $O(1)$ או $O(n)$. בלולאת while עוברים על כל הצמתים והקשתות היוצאות מהם, סה"כ n+m פעולות. בכל פעולה מוציאים את הצומת המינימלית ב-Q בסיבוכיות $\log n$. נקבל $O((n+m)\log n)$. מכיוון ש-$m=n^2$ במקרה הגרוע נסיק כי סיבוכיות אלגוריתם פרים במקרה הגרוע הוא: $O(m\log n)$.

ה) הבדלים בין אלגוריתם קרוסקל לפרים

(1) אלגוריתם קרוסקל מתחיל מיער של עצים ובכל פעם מוסיף קשת עד שהופך לעץ. פרים מתחיל מתת עץ ובכל פעם מגדיל אותו עד שהעץ כולל את כל הצמתים.

$O(m\alpha(n))=O(m)$

(2) סיבוכיות פרים טובה יותר. אך כאשר הקשתות כבר ממוינות. או שהמיון בזמן לינארי - הקרוסקל עדיף. ↑