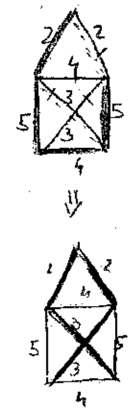
24.11.08 (1)

אלגוריתמים - שיעור 4

- עצים פורשים מינימליים (עפ"מ) Minimum Spanning Tree (MST).
הקלט הוא גרף לא מכוון וקשיר $G=(V,E)$ עם משקלות על הקשתות: לכל קשת e יש משקל $w(e)$.

- עץ פורש - תת גרף של G שמכיל את כל הצמתים והוא עץ (מורכב מ-$|V|-1$ קשתות של E).
תמיד קיים עץ כזה כי G קשיר.



משקל עץ פורש T - הוא סכום משקלי כל הקשתות של העץ הפורש. $W(T)=\sum_{e\in T} w(e)$.

בדוגמה שציירנו משקל העץ אינו מינימלי. ניתן להחליף חלק מהקשתות בקשתות אחרות "קלות" יותר, כך שמשקל העץ ירד.

מטרתנו תמיד תהיה למצוא את העץ הפורש במשקל מינימלי (עפ"מ).
עפ"מ אינו בהכרח יחיד. לדוגמה אם כל משקלות הקשתות שווים ל-1,
כל עץ פורש מינימלי הוא במשקל $|V|-1$.

מוטיבציה לדוגמה: כאשר רוצים לסלול מערכת כבישים שתחבר את כל הערים אך תהיה בעלות מינימלית. נשים לב שעץ פורש מינימלי אינו מתחשב בדרך של דומות ספציפי. כך לדוגמה, יהיו מגיעים תושבי הערים הסמוכות לסלול כביש שיחבר ביניהם. אך זה יגדיל את עלות הסלילה.

דוגמאות נוספות הן שימוש ברשת תכנון פריסת רשת טלפונים/חשמל.

הערות:

① אם כל המשקלות חיוביים, אז תת גרף קשיר מינימלי תמיד יהיה עץ. כל קשת נוספת תוסיף משקל ולכן יגדל יותר.

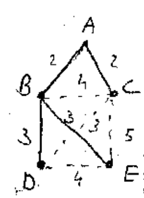
② אם יש משקלות שליליים, חשוב לדרוש עץ פורש.

③ אם מתעניינים על עץ (לא רוצים פתרון אחר), אז משקלות שליליים. ניתן להוסיף לכל המשקלות קבוע A כך שכל המשקלות יהיו חיוביים ואז המשקל של כל עץ פורש גדל בדיוק ב-$A\cdot(|V|-1)$ ולכן עפ"מ קודם הוא עפ"מ כעת.

נציג גישה כללית למציאת עפ"מ ואחר כך ניתן שני מימושים ספציפיים.

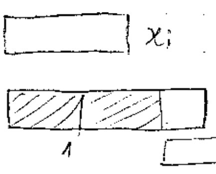
גישה חמדנית Greedy

נוסיף קשתות ונקפיד שלא ייווצר מעגל. אם נצליח להגיע ל-$|V|-1$ קשתות נקבל עץ פורש.



בחרנו בגישה החמדנית וקיבלנו עץ פורש מינימלי בגלל שזו גישה נכונה.

דוגמה לגישה חמדנית לא מוצלחת - Bin Packing:
נתונים n מספרים ממשיים $x_1, x_2, \ldots, x_n$ כך ש-$x_i \in (0,1)$ ויש מספר לא מוגבל של קופסאות (בגודל 1 - כמו משרד כדור סטים) בגודל 1.
המטרה היא לארוז את כל העצמים במספר מינימלי של קופסאות.



גישה חמדנית: נמיין את העצמים לפי הסדר בכל רגע נמצא קופסה עם מקום פנוי לעצם הבא. אם יש מקום ... לא נפתח קופסה חדשה.

דוגמה מספרית: 0.8, 0.8, 0.8, 0.8, 0.8, 0.2, 0.2, 0.2, 0.2
הפתרון האופטימלי הוא כמובן 5 קופסאות, אבל הגישה החמדנית תדרוש 6 קופסאות.

גישה כללית

<u>הגדרה:</u> תת קבוצה A של E נקראת <u>קבוצה מבטיחה</u> אם קיים עפ"מ T המכיל את A.

<u>הערה:</u> הקבוצה הריקה $\emptyset$ מבטיחה. (מוכלת בכל עפ"מ).

אם A קבוצה מבטיחה ו-e קשת שאינה ב-A, נקרא ל-e קשת בטוחה (safe) עבור A אם $A \cup \{e\}$ עדיין מבטיחה.

"אלגוריתם"

$A := \emptyset$

While A is not ST do (כל עוד מס' הקשתות ב-A קטן מ- $|V|-1$)

 מצא קשת e בטוחה עבור A

 $A := A \cup \{e\}$

Output A

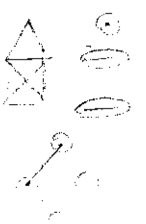
האלגוריתם נכון כי

(א) $\emptyset$ מבטיחה.

(ב) תמיד קיימת קשת בטוחה לקב' A מבטיחה שאינה עץ פורש כי A מוכלת בעפ"מ T וכל קשת e שב-T ולא ב-A היא קשת בטוחה.

(ג) לכן, גם $A \cup \{e\}$ מבטיחה.

(ד) בסוף, כש-A עץ פורש ומבטיח - A עפ"מ



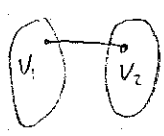
<u>הערה:</u> בקבוצה מבטיחה A אין מעגלים ולכן A יער, אוסף של עצים זרים-בצמתים (כולל עצים חסרי קשתות, עם צומת אחד)

כל רכיב קשירות בגרף (V,A) הוא עץ שיכול להיות גם ריק.

אם e קשת בטוחה ל-A היא מחברת 2 רכיבי קשירות שונים (אחרת היא תסגור מעגל) ולכן רק מאחדת אותם לרכיב קשירות אחד.

ולכן, כל צעד של האלגוריתם מקטין את מספר רכיבי הקשירות ב-1 (מאחד 2 רכיבי קשירות). בהתחלה יש $|V|$ רכיבי קשירות (כל צומת לבדו) ואחרי $|V|-1$ צעדים יש רכיב קשירות אחד - הוא העץ הפורש.

<u>חתך (cut)</u> - בגרף $G=(V,E)$ חלוקה של קבוצת הקודקודים V לשתי תתי קבוצות זרות ולא ריקות V_1, V_2 ($V_1 \cup V_2 = V$).

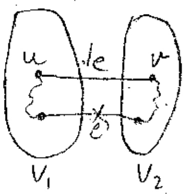


קשת e <u>חוצה</u> את החתך, אם היא מחברת צומת ב-V_1 לצומת ב-V_2.

אם A קב' מבטיחה, נאמר שהחתך <u>מכבד</u> את A אם אין ב-A קשת חוצה. $\Leftarrow$ כל רכיב קשירות של A מוכל בשלמותו ב-V_1 או ב-V_2

בהינתן חתך, הקשת e נקראת <u>חוצה קלה</u> אם משקלה מינימלי מבין כל הקשתות החוצות.

<u>טענה:</u> תהא A קבוצה מבטיחה. ויהא (V_1, V_2) חתך שמכבד את A ותהא קשת e חוצה קלה של החתך (תמיד שייכת ל- $E-A$) אזי e קשת בטוחה עבור A.



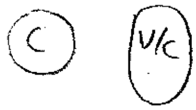
<u>הוכחה:</u> ניקח קשת חוצה קלה $e=u\cdot v$, $u\in V_1$, $v\in V_2$. לפי הנתון קיים עפ"מ T המכיל את A. אם במקרה הוא גם מכיל את e - סיימנו e בטוחה, $A\cup\{e\}$ מוכלת ב-T.

נניח לכן ש $e\notin T$. ב-T קשיר יש קשתות שחוצות את החתך. נקח ב-T מסלול π (יחיד) המחבר את u ל-v. π חייב להכיל קשת חוצה e'.
בהכרח מתקיים $w(e)\le w(e')$ כי e חוצה קלה.

נבנה עץ חדש T^* ע"י שנוציא מ-T את e' ונוסיף במקומה את e.

T^* עץ: יש בו $|V|-1$ קשתות והוא קשיר. לכן זהו עץ פורש ובנוסף ומכיוון ששיפרנו את המשקל כי $W(T^*)=W(T)-W(e')+W(e)<0$ ($W(T^*)\le W(T)$) ולכן T^* בוודאי יותר קל מעפ"מ. ולכן $W^*(T)=W(T)$ ובפרט T^* עפ"מ שמכיל את $A\cup\{e\}$, לכן e בטוחה. (בפרט $W(e')=W(e)$).

<u>מסקנה:</u> יהי C רכיב קשירות של f גז החתך $(C, V/C)$ מכבד את A ולכן קשת חוצה קלה בחתך זה היא בטוחה ל-A.



<u>אלגוריתם של Kruskal</u>

<u>הרעיון:</u> בכל שלב, נחפש קשת שמחברת 2 רכיבי קשירות שונים של A במשקל מינימלי ונוסיף אותה ל-A.

<u>המימוש:</u> כדי לממש זאת, האלגוריתם ממיין את הקשתות לפי סדר משקל עולה (חלש). ועובר עליהן בסדר זה.

לכל קשת e, נבדוק אם היא מחברת שני רכיבי קשירות שונים. אם לא, נתעלם ממנה. אם כן, נוסיף אותה ל-A ונאחד את הרכיבים לרכיב קשירות משותף.

האלגוריתם צריך לתחזק את רכיבי הקשירות של A. אלגוריתם שמתחזק אוסף תתי קבוצות זרות של V.

הפעולות הדרושות הן:

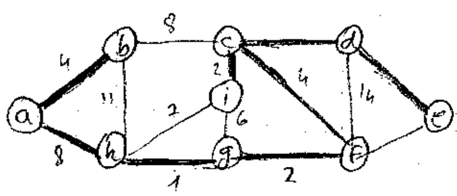
find(u) = מצא את הקבוצה המכילה את u.

Union(u,v) = איחוד את הקבוצה של u עם הקבוצה של v.

make-set(u) - יוצר סינגלטון מ-u $\{u\}$.

24.11.08 (4)

אלגוריתמים - שיעור 4

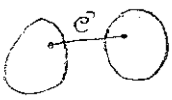


הדגמת ריצת האלגוריתם:

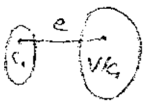
מרכיבים	סדר הבחירה
a, b, ..., ..., gh, ...	gh
a, b, ci, ..., gh, ...	ci
a, b, ci, fgh, ...	fgh
ab, cfghi, d, e	ab
ab, cfghid, e	gi - סוגרת מעגל-לא נבחר
abcfghid, e	hi - סוגרת מעגל
abcfghide	cd
	ah
	de

נכונות

מספיק להוכיח שכל קשת e שהאלגוריתם מוצא שהיא מחברת רכיבי קשירות שונים של A, היא קשת בטוחה. e היא קשת עם המשקל המינימלי המחברת 2 רכיבי קשירות שונים.



נניח שיש קשת קלה יותר מ-e עם אותה תכונה. היא נבדקה קודם ולא נבחרה.



נניח ש-e מחברת משהו ב-C_1 עם משהו ב-C_2. נבנה חתך $(C_1, V/C_1)$. נראה ש-e קשת קלה כי כל קשת קלה מחברת רכיבי קשירות שונים של A הנוכחית ו-e הכי קלה מביניהן ולכן e בטוחה לפי המסקנה הקודמת.

מימוש יעיל

<u>מיון הקשתות</u> - $O(|E|\log|V|) = O(|E|\log|E|)$ (ניתן להחליף בין V ו-E כי $|E| \le |V|^2 \Rightarrow \log|E| \le 2\log|V|$).

<u>שאר הפעולות</u> $O(|E|+|V|)$ + זמן מימוש Union Find

$2|E|$ Find כל אחת $O(\log|N|) = O(|E|\log|V|)$

$|V|-1$ Union כל אחת $O(1)$

כל תת קבוצה ממומשת ע"י עץ מכוון לפי השורש ולכן find(u) = הולכת במסלול מ-u לשורש ומחזירה את השורש בתור נציג הקבוצה. העלות ≥ עומק העץ.

Union(u,v) = נתלה את העץ של u מתחת לשורש של v או להפך. תמיד נשתול את העץ הקטן בתוך שורש העץ הגדול. אם עושים זאת, עלות Union היא $O(1)$ וזה מבטיח שעומק עץ ≥ $\log_2$ מספר הצמתים בעץ

אלגוריתם של Prim

<u>הרעיון:</u> לגדל רק רכיב קשירות אחד "אמיתי" של A וכל השאר נשארים טריוויאליים.

אלגוריתמים - שיעור 4

בכל שלב צריך למצוא צומת שעדיין לא מחובר עם קשת במשקל מינימלי שמחברת אותו לרכיב הנבנה ולכן לכל צומת כזה יהיה משקל השווה למשקל הקשת המחברת הקלה ביותר. ובכל שלב נחפש את הצומת הקל ביותר שמחובר לרכיב הקשירות הנבנה