

עצים פורשים מינימליים

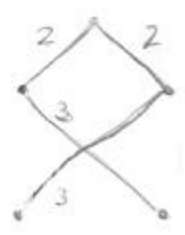
קלט: זרע G לא מכוון וקשר עם משקלות על הקשתות: לכל קשת e יש משקל $w(e)$.

הצורה: על פורש של זרע G הוא תת-זרע של G שמכיל את כל צמתיו G , והוא על. תמיד קיים כזה, כי G קשיר.



הצורה: משקל של על פורש T הוא סכום משקלות הקשתות ב- T .

צומת: הצל הפורש בעל המשקל המינימלי בזרע לעיל הוא:



ומשקלו 10.

מראה: בהינתן הקלט שהצגנו, למצוא על פורש בעל משקל מינימלי (על זה אינו בהכרח יחיד, למשל, אם כל משקלות הקשתות שווים ל-1, כל על פורש הוא עפ"מ, שמקלו $|V|-1$).

הערה: אם כל המשקלות חיוביים, אז תת-זרע קשיר מינימלי יהיה בהכרח על. אם יש משקלות שליליים, חשוב לזרוע על פורש.

אם מתקשה על על, אז משקלת שליליים אינם משמעותיים, כי אינו יכולים להוסיף למשקל כל קשת. סכום A גדול, ואז המשקל של כל על פורש גדול בדיוק ב- $A(|V|-1)$. לכן על שזה עפ"מ קודם, "שאר עפ"מ", ולהיפך.

נציג שיהיה כללי למציאת עפ"מ, ואח"כ ניקח שני יישומים קונקרטיים. הנישה שנציג תהיה גישה חמדנית (Greedy).

כלל, גישה חמדנית לא מבטחת תוצאה אופטימלית. בדוגמה, נביא את בעיית Bin Packing:

נתונים מספרים ממשיים בין 0 ל-1 (בקטע הפתוח). יש מספר לא מוגבל של קובסאות, שכל אחת מהן היא כמו שלל של בדורי ג'ייס - כמות, יש לה מידת אחוז המורגל: אורך. כל קובסה היא באורך 1. המטרה היא לארוז את העצמים (המספרים) במספר מינימלי של קובסאות. גישה חמדנית: נארוז את העצמים לפי הסדר. כל יחס נחשב קובסה אם מקום

סנוי לעצמ הקא. אם יש נשים אלו, אם לא, נשיא אלו בקופסה חדשה.
 נראה שהשיטה לא אובליגטורית, באמצעות דוגמה:

0.2, 0.2, 0.2, 0.2, 0.2, 0.8, 0.8, 0.8, 0.8, 0.8

מחבר כי האלגוריתם האובליגטורי תהיה ב-5 קופסאות, אך האלגוריתם המחבר ידרוש 6 קופסאות.

(הקורא היציג יחשוב דווקא, שסנוי קם האלגוריתם יבא לפתרון אובליגטורי.
 למשל, לעבור על העצמים מהשדה לקטן, ולא לפי הסדר. ערוץ המכל, גם זה
 לא פתרון אובליגטורי. למעשה, לא יבוצע פתרון אובליגטורי במסגרת ריצה
 פולינומית, ויש סתירה עבורה שאין ככה, משום שהבעיה היא NP-שלמה.
 באופן מסתעף, גישה חמורה למציאת ענ"מ דווקא עובדת, כפי שרואה בהמשך.

פסדו: תת-קבוצה A של E נקראת קבוצה מבטיחה אם קיים ענ"מ T
 המכיל את A. הערה: הקבוצה הריקה $\emptyset$ מבטיחה.

אם A קבוצה מבטיחה, ו- $e \notin A$ קשה, נקרא לה קשה באותה
 עבור A אם $A \cup \{e\}$ עדין מבטיחה.

אלגוריתם:

$A := \emptyset$

while A is not MST ($\Leftrightarrow |V|-1 > A$)

find e which is safe for A (נכון, לא ברור אם עדיין זה)

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

output A

האלגוריתם נכון כי: א) $\emptyset$ מבטיחה.

ב) המיד קיים קשה באותה קבוצה מבטיחה A שאיננה קשה סוף,
 כי A מוכלת בענ"מ T, וכל קשה e ב T ולא ב-A
 היא באותה.

לכן אם $A \cup \{e\}$ מבטיחה, בסוף A קשה פורש ומבטיח, ולכן
 הוא ענ"מ.

הערה: בקבוצה מבטיחה A אין העצמים, ולכן A ישר - אולם של עצמים
 במחברים, כולל עצמים חסרי קשרים עם צמת אחר.

כל ריבוי קשרים בעל (V,A) יהיה על שכול להיות זה חלקי אם e
 קשה באותה A-הוא מחברת 2 ריבוי קשרים שונים (אחרת הוא עצום
 מעל) ועל כן מאחדת אתם לריבוי קשרים אחד. לכן כל עצם של
 האלגוריתם מקיף את מספר ריבוי קשרים ב-1.
 בהימנעות יש ואל ריבוי קשרים, ואחר (1-1) עצמים יש ריבוי קשרים
 אחד, והוא קשה הפורש.

הצגה: חתך (cut) בקרן $G=(V,E)$ הוא פירוק של קבוצת הקווקזים V לשתי קבוצות זרות ולא ריקות V_1, V_2 .
 קשת e חוצה את החתך, אם היא מחברת נחמה מ- V_1 לבחנה מ- V_2 .
 אם A קבוצת מכונה, נאמר שהחתך מכבד את A אם אין ב- A קשת חוצה.
 בהינתן חתך, קשת נקראת חוצה קלה אם יש לה משקל מינימלי מבין הקשתות החוצות.

לענה: יהא A קבוצת מכונה, ויהא (v_1, v_2) חתך שמכבד את A , יהא e קשת חוצה ^{קלה} של החתך. אז e קשת במחנה עבור A .

הוכחה

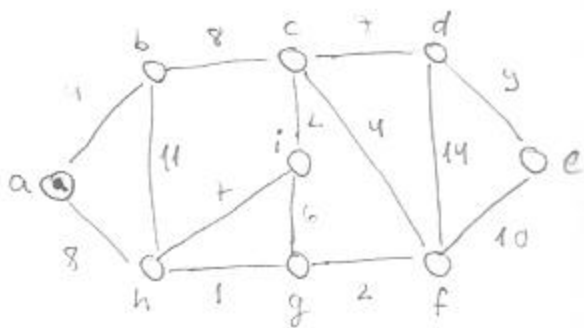
ניקח קשת חוצה קלה $e=(u,v)$ ($u \in V_1, v \in V_2$). לפי הנתון קיים ענ"ח T המכיל את A . אם במקרה זהו גם מכיל את e , $T \supseteq A \cup \{e\}$, ולכן ברור $e \in T$.

נניח כי $e \notin T$. ב- T יש בהכרח קשתות שחוצות את החתך.
 ניקח את המסלול היחיד ב- T שמחבר בין u ל- v , ונקרא לו π . π כולל
 לפחות קשת חוצה e' . ברור כי $w(e) \leq w(e')$.

במקרה של חציה T^* ע"י מה שנתנו את e' וניסוף במקומו את e .
 (זה אכן ע"י, כי יש בו $|V|-1$ קשתות, והוא קשיר).
 מתקיים: $w(T^*) = w(T) - w(e') + w(e) \leq w(T) \Rightarrow w(T^*) = w(T)$

מכאן $e \in T^*$ הוא של פורש מינימלי. בסוף, קיים ענ"ח שמכיל את $A \cup \{e\}$,
 ולכן e במחנה. $\square$

השערה המפורסמת נקראת ענ"ח מינימלי כוללת את כל קשתות החתך שמכבדות את A .



הוכחה: יהא e רכיב קשיר של A , sk החתך $(e, V \setminus e)$ שמכבד את A , ולכן קשת חוצה קלה בחתך זה, במחנה ל- A .

האלגוריתם של Kruskal (מפורסם במיוחד)

הנניח: בכל שלב, נחפש קשת שמחברת 2 רכיבי קשירים שונים של A במשקל מינימלי, וניסוף אותה ל- A .

בכדי להבטיח זאת, האלגוריתם מניח את הקשתות לפי סדר משקל עולה (תחלה) ודוחק

הסדר זה, לכל קשת e נבדוק אם היא מחברת שני רכיבי קשיות שונים, ואם היא לא נחשבת מחברת. אם היא כן, נוסף אליה A ונמחק את הרכיבים הרכיבים קשיות משותף.

הפעולה המרכזית במימוש שתארנו היא לבדוק האם e מחברת רכיבי קשיות, ואם כן ניתן למצוא ביצירה e מחצית רכיבי קשיות. $union-find$.

דוגמה הרכבה (על הסל שבצד שמאל): הקשת הראשונה $efgh$ מחברת ef . אולי נוסף את ef ואז bc , בשלב זה, מצב $union-find$ יהיה:

a, b, c, d, e, f, g, h

אחרי נוסף את cd ואז ab , עכשיו נבדוק אם hi היא לא נוסף אליה כי קודם לא מחברת רכיבי קשיות.

נבדוק את de ונבדוק נוסף אליה. לאחר מכן נוסף את ah ואז את de , וזה כבר ניתן לנו ע"מ.

(אם אתם לא סבורים על מה שקרה פה, עשו הדגמה חיצונית מפורטת).

כאונות מסבין להוכיח שכל קשת e שהיא צמודה מוצא, שהיא מחברת רכיבי קשיות שונים A , הנה קשת באותה.

e היא קשת עם המסלול המינימלי המחברת שני רכיבי קשיות שונים:

נניח שיש קשת קלה יותר e' עם אותו גודל. אזי e' נבדקה קודם ולא נכנסה.

נניח $e-e'$ מחברת משהו B עם C (נניח C חסר C_1, C_2).

הבדל $e-e'$ קשת חוצה קלה, כי כל קשת חוצה מחברת רכיבי קשיות שונים על A הרכיבים, ו- e הרי קלה מבין הרכיבים. אז e באותה, לפי האסטרטגיה המרכזית.

מחצית חוצה:

* מיון בקשתות: $O(|E| \cdot \log |E|) = O(|E| \cdot \log |V|)$

(המספר נכון כן: $|E| \leq |V|^2 \Rightarrow \log |E| \leq 2 \cdot \log |V|$)

* $2|E|$ פעולות: FIND

* $|V|-1$ פעולות: UNION

$O(|V|)$

(נניח את $union-find$ כק: כל מחצית קידומת מאומתת ע"י של מחצית

בא"י השני. FIND יוצר את שדה ריבוי הקשיות. ברור כי שדה

$FIND \geq$ אורך השדה. את $union$ נניח על ידי הרכיבים השדה הקרן מחברת

שדה השדה השדה.

אם כן שדה שדה שדה, UNION מוכן $O(1)$, ומכאן שדה כן

שדה $\geq$ $\log |V|$ וכן שדה FIND היא $O(\log |V|)$.

* פעולות נוספות, לא צריכות. $\Leftarrow$ סה"כ $O(|E| \cdot \log |V|)$

הדיון: לכל קו רכיב קטנות אחד "אמתי" של A, בשאר הקונקורס (שארם singletons).

בכל שלב צריך למצוא צומת שצריך לא לחזור עם קטת במסלול
מינימלי שמחבר אותו לרכיב הנבחר. לכן לכל צומת כזה יהיה
מסלול השווה למסלול הקטת המחברת הקלה ביותר.
בכל שלב נחפש את הצומת הקלה ביותר שיצטרף לרכיב הקטנות
הנבחרת.