הרצאה 4

עץ פורש מינימלי

הגדרות ואבחנות

יער - גרף חסר מעגלים

עץ - יער קשיר

חתך - תת קבוצה של צמתים $S \subseteq V$

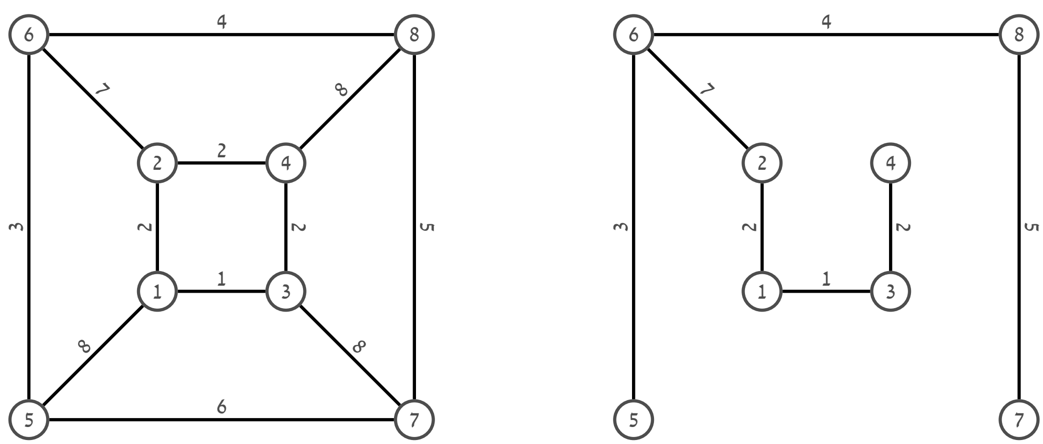
קשת חוצה - קשת uv חוצה חתך S אם $|\{u,v\} \cap S| = 1$

אבחנה 1. הוספת קשת uv לעץ סוגרת מעגל שמכיל את הקשת + המסלול מ-v ל-u בעץ. כעת ניתן להסיר כל קשת מהמעגל הנ"ל ולקבל בחזרה עץ.

אבחנה 2. אם קשת uv נמצאת על מעגל והיא חוצה חתך S אז את S חוצה קשת אחת נוספת (לפחות) ששייכת למעגל.

הגדרה 1 (עץ פורש מינימלי)**.** בהינתן זוג של גרף לא מכוון $G = (V,E)$ ופונקציית משקל (אי שלילית) $w : E \to \mathbb{R}$ עץ פורש מינימלי הוא כל עץ $T = (V,F)$ שממזער את הערך $\sum_{e \in F} w(e)$.

דוגמה



הכלל הכחול

בחלק זה נראה אלגוריתם גנרי שפועל על פי הכלל הכחול.

הגדרה 2 (קשת קלה)**.** בהינתן חתך, S, ופונקציית משקל, $w : E \to \mathbb{R}$, קשת e שחוצה את S תקרא קלה אם לא קיימת קשת אחרת שחוצה את S, e' שמקיימת $w(e') < w(e)$.

1. אתחול: $B \leftarrow \emptyset$ (קשתות כחולות)
2. כל עוד $T = (V,B)$ אינו קשיר הפעל את הכלל הכחול:
   (א) בחר חתך לבן, S, וקשת קלה, e, שחוצה אותו
   (ב) $B \leftarrow B \cup \{e\}$

טענה 1. האלגוריתם מחזיר עץ

הוכחה. קשירות נובעת מיידית מהגדרת האלגוריתם. נניח בשלילה שנסגר מעגל. נסתכל על הנקודה בה האלגוריתם צובע את הקשת שסוגרת את המעגל הכחול. לפי אבחנה 2 קיימת קשת כחולה נוספת שחוצה את החתך - סתירה. □

טענה 2. האלגוריתם מחזיר עץ פורש מינימלי

הוכחה. נוכיח באינדוקציה שבכל שלב בריצת האלגוריתם אוסף הקשתות הכחולות מוכל בתוך עץ פורש מינימלי כלשהו:
בסיס: טריוויאלי באתחול
צעד: לפי ההנחה קיים עפ"מ שמכיל את i הקשתות הכחולות הראשונות. נניח שהקשת, e, שהוספנו בשלב ה-$i+1$ לא שייכת לעפ"מ הנ"ל (אחרת סיימנו). נוסיף את e לעפ"מ, סגרנו מעגל. במעגל זה קיימת קשת, $e' \neq e$, שחוצה את S, החתך (הלבן) שגרם להוספת e. לפי הגדרת האלגוריתם $w(e) \leq w(e')$ ולכן ניתן להחליף בין הקשתות הנ"ל ולקבל עפ"מ שמכיל גם את e. □

הכלל האדום

בחלק זה נראה אלגוריתם גנרי שפועל על פי הכלל האדום.

הגדרה 3 (קשת כבדה)**.** בהינתן מעגל, C, ופונקציית משקל, $w : E \to \mathbb{R}$, קשת e במעגל C תקרא כבדה אם לא קיימת קשת אחרת במעגל C, e' שמקיימת $w(e') > w(e)$.

1. אתחול: $R \leftarrow \emptyset$ (קשתות אדומות)
2. כל עוד ב-$T = (V, E \setminus R)$ יש מעגל הפעל את הכלל האדום:
   (א) בחר מעגל לבן, C, וקשת כבדה, e, על המעגל
   (ב) $R \leftarrow R \cup \{e\}$

הערה: במקום לצבוע את הקשת באדום ניתן פשוט למחוק אותה מהגרף

טענה 3. *האלגוריתם מחזיר עץ*

הוכחה. חוסר מעגלים נובע מיידית מהגדרת האלגוריתם. נניח בשלילה שאלגוריתם פוגע בקשירות, כלומר קיים חתך כך שהאלגוריתם צובע באדום את כל הקשתות שחוצות אותו, נסתכל על הקשת האחרונה שנצבעה באדום ועל המעגל הלבן שגרם לה להיבחר לפי אבחנה 2 קיימת קשת לבנה נוספת שחוצה את החתך - סתירה. □

טענה 4. *האלגוריתם מחזיר עץ פורש מינימלי*

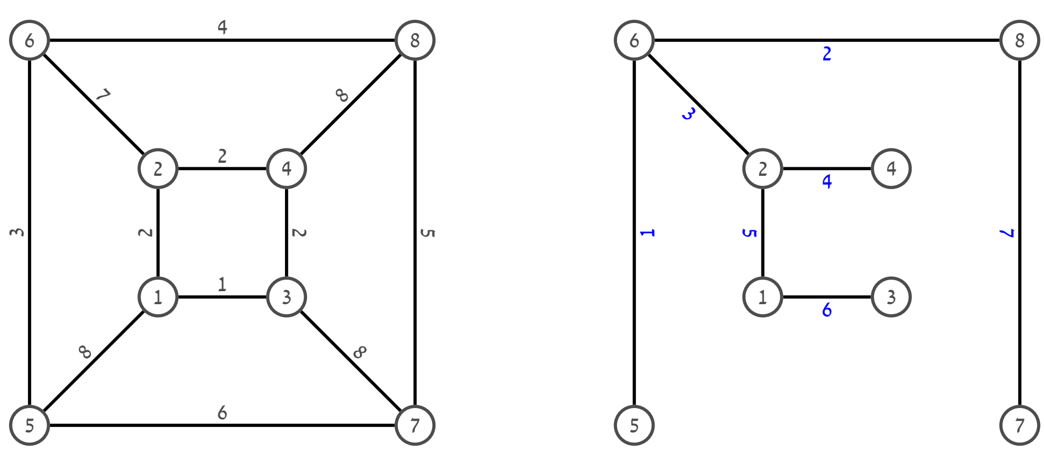
הוכחה. נוכיח באינדוקציה שבכל שלב בריצת האלגוריתם אוסף הקשתות הלבנות מכילות עץ פורש מינימלי כלשהו:
בסיס: טריוויאלי באתחול
צעד: לפי ההנחה אוסף הקשתות הלבנות בצעד ה-i מכילות עפ"מ כלשהו. נניח שהקשת, e, שמחקנו בשלב ה-$i+1$ שייכת לעפ"מ הנ"ל (אחרת סיימנו). נסתכל על החתך שמחיקת הקשת יוצרת, ועל המעגל (הלבן) שגרם למחיקת הקשת, אז קיימת קשת נוספת על המעגל שחוצה את החתך, $e' \neq e$ כך ש-$w(e) \geq w(e')$ ולכן ניתן להחליף בין הקשתות הנ"ל כך שהעפ"מ החדש מוכל כולו בקשתות לבנות. □

כחול + אדום

נשים לב שבהוכחה של טענה 2 הוכחנו שאם בשלב מסויים אוסף הקשתות הכחולות מוכלות בעמ"פ אז ניתן להפעיל את הכלל הכחול ולשמר את הטענה. באותו אופן, בהוכחה של טענה 4 הוכחנו שאם בשלב מסויים אוסף הקשתות שאינן אדומות מכילות עמ"פ אז ניתן להפעיל את הכלל האדום ולשמר את הטענה. נובע מכך שניתן להפעיל סדרה של כללים כחולים ואדומים בסדר שרירותי ולשמר את שתי הטענות.

אלגוריתם פרים Prim

אלגוריתם פרים מתחיל מעץ שמכיל צומת אחד ובכל איטרציה מוסיף לעץ את הקשת הכי זולה שקצה אחד שלה בדיוק נוגע בעץ. דוגמה:



פורמלית:

1. אתחול: $U \leftarrow \{u\}, B \leftarrow \emptyset$, כאשר u צומת שרירותי.
2. כל עוד $U \neq V$
הפעל את הכלל הכחול על החתך U ועדכן את B

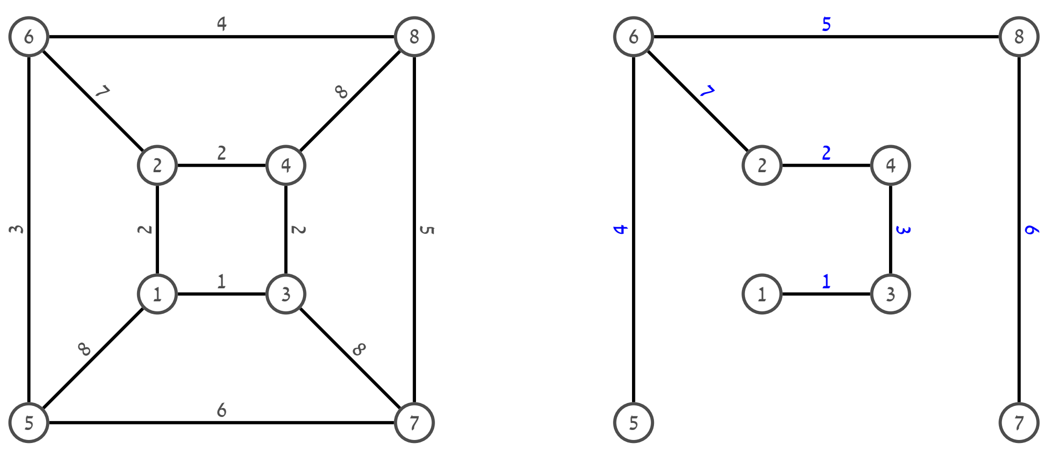
אבחנה 3. אלגוריתם פרים הוא מימוש של האלגוריתם הכללי שמפעיל את הכלל הכחול.

הערות:

- כדי לממש את האלגוריתם צריך לדעת בכל שלב אילו קשתות חוצות את החתך ולבחור מהן את הקלה ביותר
- כאשר מוסיפים צומת לחתך יתכנו השינויים הבאים:
  - קשת שחצתה את החתך עכשיו היא פנימית לחתך
  - קשת חיצונית לחתך עכשיו חוצה את החתך
- השינויים היחידים הם עבור קשתות שנוגעות בצומת שהוסף לחתך ולכן בכל פעם שמוסיפים צומת לחתך מעדכנים רק את הקשתות שנוגעות בו, סך הכל מעדכנים כל קשת פעמיים לכל היותר.
- נשים לב שאם שומרים את הקשתות שחוצות את החתך בערימת מינימום אז אנחנו מבצעים $|E|$ הכנסות ו-$|V|$ הוצאות. הוצאה והכנסה של כל קשת לוקחת $O(\log |E|)$ לכל היותר.
- סך הכל זמן הריצה של האלגוריתם הוא $O(|E| \log |E|) = O(|E| \log |V|)$
- קיימים מבני נתונים יותר יעילים עם פונקציונליות של ערימת מינימום שתומכים בהכנסה בזמן ממוצע $O(1)$ והוצאה בזמן $O(\log |E|)$ ולכן ניתן לממש את האלגוריתם בזמן $O(|E| + |V| \log |V|)$

אלגוריתם קרוסקל Kruskal

אלגוריתם קרוסקל מתחיל מיער ללא קשתות, בכל שלב באלגוריתם נמזג שני רכיבי קשירות באמצעות הקשת הקלה ביותר שמחברת שני רכיבי קשירות.
דוגמה:



פורמלית:

1. אתחול: $C \leftarrow \{\{v\} : v \in V\}$,
2. כל עוד $|C| > 1$
בחר את הקשת עם המשקל המינימלי שמחברת שני רכיבי קשירות, הפעל עליה את הכלל הכחול ועדכן את C בהתאם

הערות:

- ניתן לממש את האלגוריתם באופן הבא:
  - מיין את הקשתות בסדר לא יורד של משקלן

- – עבור כל קשת לפי הסדר, אם היא מחברת שני רכיבי קשירות הפעל עליה את הכלל הכחול ועדכן את C

- זמן הריצה של מימוש כזה הוא $O(|E|\log|E|)$ עבור המיון ובנוסף לכל קשת צריך לבדוק אם היא מחברת שני רכיבים שונים ואם כן לעדכן את מבנה הרכיבים. כזכור מקורס מבני נתונים קיים מימוש פשוט שעושה זאת בזמן ממוצע של $O(\log|V|)$ ולכן הזמן הריצה הכולל הוא $O(|E|\log|V|) = O(|E|\log|E|)$.