אלגוריתמים 1

תוכן העניינים

פרק 1

אלגוריתמי BFS ו-DFS

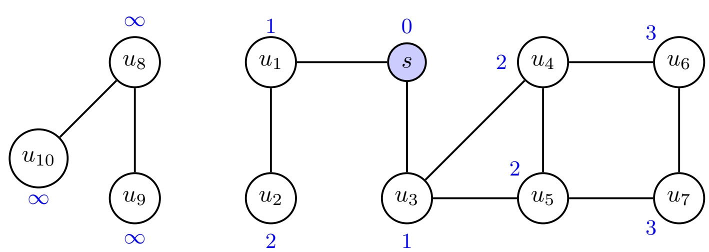
1. BFS - Breadth First Search

שאלה 1.1 כיצד לחשב מסלול קצר ביותר בין שני צמתים בגרף לא מכוון G?

1.1. הגדרת המרחק בגרף לא מכוון.

הגדרה 1.1 (המרחק בין צמתים u, v בגרף G)
בהינתן גרף לא מכוון $G = (V, E)$ ושתי צמתים $u, v \in V$,

המרחק בין u ו-v ב-G הוא האורך (מספר קשתות) של המסלול הקצר ביותר בין u ו-v ב-G. נסמן מרחק זה ב-$\delta_G(u, v)$ או ב-$\delta(u, v)$.



איור 1: המרחקים $\delta(s, u)$ מצומת s בגרף לא מכוון זה מסומנים בכחול ליד כל צומת $u \in V$.

טענה 1.1 (המקבילה לאי-שוויון המשולש)
יהי $G = (V, E)$ גרף לא מכוון, ויהי $s \in V$. לכל קשת $e = (u, v) \in E$ מתקיים:

$$\underbrace{\delta(s, v)}_{\text{המרחק בין } s \text{ ו-} v} \leq \underbrace{\delta(s, u)}_{\text{המרחק בין } s \text{ ו-} u} + \underbrace{1}_{\text{אורך הקשת } e}$$

הוכחת הטענה. אם אין מסלול בין u ו- s ב-G אז $\delta(s, u) = \infty$ והטענה מתקיימת.

אחרת, יהי P מסלול קצר ביותר בין s ו-u ב- G, כאשר אורכו שווה ל-$\delta(s, u)$.
נשרשר ל-P את הקשת e, וקיבלנו מסלול ב-G בין s ו-v, שאורכו שווה ל-$\delta(s,) + 1$.

$\delta(s, v)$ שווה לאורך המסלול הקצר ביותר בין s ו-v ב-G, ולכן $\delta(s, v) \leq \delta(s, u) + 1$. ■

1.2. מוטיבציה לאלגוריתם BFS. נרצה לחשב את המרחק בין צומת s לכל צומת בגרף:

- קלט: גרף לא מכוון $G = (V, E)$ וצומת $s \in V$.
- מטרה: לחשב לכל $v \in V$ את $\delta_G(s, v)$.

האינטואיציה: להתחיל מהצומת היחיד שעבורו יודעים את $\delta(s,?)$, וזה s עצמו.

1.3. אלגוריתם ה-BFS.

- אתחול: $\lambda(v) \leftarrow \begin{cases} 0 & v=s \\ \infty & v\neq s \end{cases}$, $T \leftarrow \{s\}$, $Q \leftarrow \{s\}$ תור.
- כל עוד $Q \neq \emptyset$:
  (1) יהי u הצומת בראש התור Q.
  (2) לכל קשת $e=(u,v)\in E$ כך ש-$v\notin T$:
    (א) $T \leftarrow T \cup \{v\}$.
    (ב) $\lambda(v) \leftarrow \lambda(u)+1$.
    (ג) הכנס את v לסוף התור Q.
  (3) הוצא את u מהתור Q.

1.4. נכונות האלגוריתם.

הערה 1.1 (סימון מקובל בקורס) עבור גרף $G=(V,E)$, נסמן $|V|=n,\ |E|=m$.

שאלה 1.2

(1) מדוע האלגוריתם מחזיר תשובה נכונה?
(2) עד כמה האלגוריתם יעיל? (בד"כ יעילות תתייחס לזמן)

<u>נתחיל מ-(2)</u>.

- האתחול: $O(n)$.
- האיטרציה בה u יוצא מ-Q: $O(\deg(u))$.

כמו כן,

- כל צומת נכנס ל-Q לכל היותר פעם אחת.
- כל צומת שנכנס ל-Q גם יוצא מ-Q.

סך הכל זמן ריצה:

$$\underbrace{O(n)}_{\text{אתחול}} + \underbrace{O\left(\sum_{u\in V}\deg(u)\right)}_{\substack{\text{חסם עליון על}\\ \text{זמן הריצה של}\\ \text{כל האיטרציות}}} = O(n+m)$$

■

נתמקד בטענה (1), ונוכיח אותה תוך שימוש בטענות העזר הבאות:

למה 1.1 ("λ לא מפספס למטה") יהי $G=(V,E)$ גרף לא מכוון, ותהא צומת $s\in V$. יהיו $\forall v\in V,\ \lambda(v)$ הסימונים שהתקבלו מריצת BFS על G החל מ-s. אזי:

$$\lambda(v) \geq \delta(s,v)\,,\ \forall v\in V$$

הוכחה. יהי $v \in V$.

אם v לא נכנס ל-Q, יתקיים $\lambda(v) = \infty$ והטענה נכונה.

אם v נכנס ל-Q (וזה קורה בדיוק פעם אחת), נוכיח את הטענה באינדוקציה על סדר כניסת הצמתים ל-Q:

- בסיס: s נכנס ראשון לתור (המקרה ש-$v = s$), ואז:

$$\lambda(s) = \underbrace{0}_{\text{הגדרת האלגוריתם}} = \delta(s,s)$$

- צעד: נניח נכונות עבור k הצמתים הראשונים שהוכנסו לתור, נניח כי v היא הצומת ה-$k+1$ שהוכנסה לתור.

ברגע ההכנסה של v ל-Q, נסמן ב-u את הצומת שבראש Q, ונקבל:

$$\lambda(v) \underbrace{=}_{\text{הגדרת האלגוריתם}} \lambda(u)+1 \underbrace{\geq}_{\substack{\text{הנחת אינדוקציה} \\ u \text{ עבור}}} \delta(s,u)+1 \underbrace{\geq}_{\substack{\text{אי שוויון המשולש} \\ (u,v)\in E\text{-ו } s \text{ עבור}}} \delta(s,v)$$

■

למה 1.2 יהי $(v_1, v_2, \ldots, v_k)$ תוכן Q בשלב כלשהו של ריצת BFS על G החל מ-s. אזי:

(1) $\lambda(v_1) \leq \lambda(v_2) \leq \ldots \leq \lambda(v_k)$.

(2) $\lambda(v_k) \leq \lambda(v_1) + 1$.

הוכחה. נוכיח באינדוקציה על סדר הפעולות של הכנסה/הוצאה מ-Q:

- בסיס: האתחול הוא כש-Q מכיל רק את s. לכן (1) ו-(2) מתקיימים באופן ריק.
- צעד: נניח נכונות עבור r הפעולות הראשונות, ונוכיח עבור הפעולה ה-$r+1$.

אם הפעולה ה-$r+1$ הייתה הכנסה, נניח שהכנסנו את v ו-u בראש התור, אז:

$$\lambda(v) = \lambda(u) + 1$$

לפי הגדרת האלגוריתם.

בגלל שלפני הוספת v ל-Q (1) ו-(2) התקיימו, זה יתקיים גם לאחר הוספת v.

אם ההפעלה ה-$r+1$ הייתה הוצאה, אז ברור שמהנחת האינדוקציה (1) ו-(2) יתקיימו גם לאחריה.

■

משפט 1.1 (הוכחת נכונות אלגוריתם BFS)

יהי $G = (V,E)$ גרף לא מכוון ו-$s \in V$.

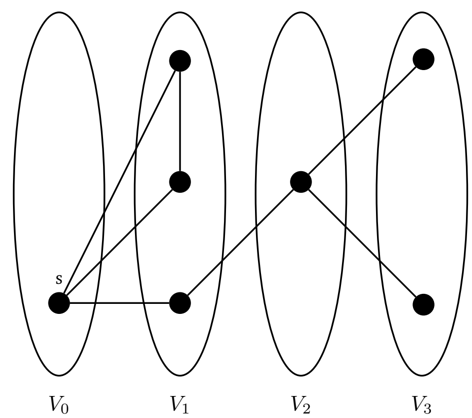
אז בסיום ריצת BFS על G החל מ-s מתקיים:

$$\forall v \in V,\ \lambda(v) = \delta(s,v)$$

הוכחת המשפט משתמשת בטענות 1 ו-2.

רעיון ההוכחה: נסתכל על שכבות הגרף לפי מרחקן מ-s:

$$V_k \triangleq \{u \in V : \delta(s,u) = k\}$$



איור 2: שכבות של גרף לא מכוון לדוגמה עבור צומת s כלשהי

<u>הוכחת המשפט</u>. נניח שב-G אין מסלול בין s ו- v $\Longleftarrow$ $\delta(s,v) = \infty$.
לפי טענה 1 נקבל ש-$\lambda(v) \geq \infty$, כלומר $\lambda(v) = \infty$, והמשפט נכון.

נניח שב-G יש מסלול בין s ו-v ונסמן $\delta(s,v) = k$.
נוכיח את המשפט באינדוקציה על k:

- <u>בסיס</u>: $k = 0$, אז $v = s$, והמשפט מתקיים מפני שבאתחול מוגדר $\lambda(s) = 0$.
- <u>צעד</u>: נניח כי $v \in V_k$, ונסמן:

$$A \triangleq \{u \in V_{k-1} | (u,v) \in E\}$$

כאשר הגדרת A אינה תלויה באלגוריתם.
נסמן ב-u^* את הצומת ב-A שהיא <u>הראשונה</u> לצאת מהתור Q.

נשים לב ש-A אינה יכולה להיות ריקה, ולפי הנחת האינדוקציה,
בסיום ריצת האלגוריתם לכל הצמתים ב-A ישנו ערך λ השווה ל-$k-1$,
ולכן בהכרח כל אחד מהם הוכנס לתור Q.

נראה שבאיטרציה שבה u^* נמצא בראש התור Q, לצומת v מתקיים ש-$\lambda(v) = \infty$ (כלומר, v עדיין "לא התגלה").

נניח בשלילה שזה לא המצב, ולכן יש איטרציה קודמת לזו ש-u^* נמצא בה בראש Q, שבה v מוכנס לתור Q (ונניח ש-w נמצא בראש התור Q באיטרציה זו).

בגלל בחירת u^*, מתקיים ש-w הוא שכן של v בשכבה j, כך ש-$0 \leq j \leq k-1$ (נובע מלמה 1.2).

לפי הנחת האינדוקציה $\lambda\left(w\right)<\lambda\left(u^{*}\right)$, וכעת:

$$\lambda\left(v\right)\underbrace{=}_{\text{הגדרת האלגוריתם}}\lambda\left(w\right)+1<\lambda\left(u^{*}\right)+1\underbrace{=}_{\substack{\text{הנחת האינדוקציה}\\ u^{*}\ \text{עבור}}}\left(k-1\right)+1=k=\delta\left(s,v\right)$$

סה"כ קיבלנו $\lambda\left(v\right)<\delta\left(s,v\right)$, וזו סתירה מלמה 1.1.

באיטרציה שבה u^{*} בראש התור Q, הצומת v מקיימת $\lambda\left(v\right)=\infty$, ולכן באיטרציה זו v יקבל סימון $\lambda\left(v\right)=k$ ויוכנס ל-Q.

■

2. DFS - Depth First Search

משימה: למצוא רכיבים קשירים היטב של גרף מכוון בזמן לינארי.

2.1. חותמות זמן: זמני גילוי וסיום של צומת (במהלך אלגוריתם סריקה).

הגדרה 1.2 $s(u)$ - זמן הגילוי של צומת u

הגדרה 1.3 $f(u)$ - זמן סיום של צומת u.

2.2. האלגוריתם.

- אתחול:

(1) $\forall u \in V,\ \text{status}(u) \leftarrow \text{unvisited}$

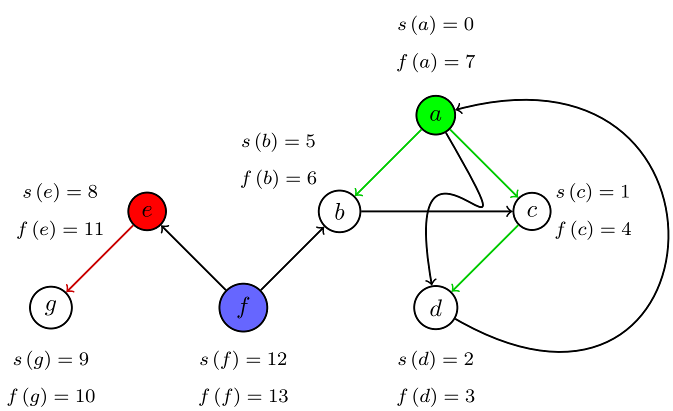
(2) $\forall u \in V,\ \begin{matrix} p(u) \leftarrow \text{NULL} \\ t \leftarrow 0 \end{matrix}$

- כל עוד יש צומת u כך ש-$\text{status}(u) = \text{unvisited}$: בצע $\text{visit}(u)$.
- $\text{visit}(u)$:

(1)
- $s(u) \leftarrow t$
- $t \leftarrow t + 1$
- $\text{status}(u) \leftarrow \text{visited}$

(2) לכל קשת $(u \to v) \in E$, אם $\text{status}(v) = \text{unvisited}$, אז $p(v) \leftarrow u$, וגם $\text{visit}(v)$.

(3) $\begin{cases} f(u) \leftarrow t \\ t \leftarrow t + 1 \end{cases}$



איור 3: דוגמת הרצה של אלגוריתם DFS.

מסקנה 1.1 בריצת DFS על גרף מכוון G, לכל צומת $u \in V$,
$\text{visit}(u)$ יקרא בדיוק פעם אחת.

2.3. זמן ריצה.

- מה זמן הריצה של אלגוריתם ה-DFS?
- עבור צומת $u \in V$, כמה זמן לוקח לבצע $\text{visit}(u)$ ללא הקריאות הרקורסיביות (אם יש) שנובעות ממנו? - $O(1)+O(\deg_{\text{out}}(u))$.

$\Longleftarrow$ סה"כ $O(n+m)$ (ובפרט האלגוריתם עוצר).

הערה 1.2 לאלגוריתם ה-DFS דרגות חופש רבות.
חותמות הזמן s, f מהוות תיעוד של היסטוריית ריצת האלגוריתם.

הגדרה 1.4 (יער ה-DFS) נסתכל על הגרף $G_p=(V,E_p)$, כאשר:

$$E_p=\{(p(v)\to v)\in E : p(v)\neq \text{NULL}\}$$

נשים לב ש-G_p הוא תת-גרף של G.

משפט 1.2 (תרגיל) G_p הוא יער מכוון אשר פורש את כל צמתי V.

2.4. סוגי קשתות ביער ה-DFS.

שאלה 1.3 כיצד ניתן לסווג את קשתות G
בהינתן ריצה מסוימת של DFS?

הגדרה 1.5 (קשת עץ) $(u\to v)\in E$ היא קשת עץ, אם $p(v)=u$.

הגדרה 1.6 (קשת קדמית) $(u\to v)\in E$ היא קשת קדמית, אם אינה קשת עץ,
ובנוסף u אב קדמון של v ביער ה-DFS.

הגדרה 1.7 (קשת אחורית) $(u\to v)\in E$ היא קשת אחורית, אם u צאצא של v ביער ה-DFS.

הגדרה 1.8 (קשת חוצה) כל שאר הקשתות מכונות קשתות חוצות.

הערה 1.3 כאשר מבצעים DFS על גרף לא מכוון, יווצרו רק קשתות עץ וקשתות אחוריות (ללא הוכחה).

2.5. אפיון יחסי אב-צאצא ביער ה-DFS.

למה 1.3 לכל גרף מכוון G, לכל ריצת DFS ולכל $u,v\in V$,
בדיוק אחד משלושת הבאים מתקיים:

(1) $[s(u),f(u)]$ ו-$[s(v),f(v)]$ זרים, ו-u אינו צאצא של v ו-v אינו צאצא של u.
(2) $s(v)<s(u)<f(u)<f(v)$, ו-u צאצא של v.
(3) $s(u)<s(v)<f(v)<f(u)$, ו-v צאצא של u.

הוכחה. נניח $s(u)<s(v)$ (המקרה ההפוך - סימטרי).

- מקרה ראשון: $\boxed{s(v)<f(u)}$

נרצה להראות שאנחנו במקרה ג'.

ברגע גילוי v, עדיין לא סיימנו את $\text{visit}\left(u\right)$ (בגלל ש-$s\left(v\right)<f\left(u\right)$).

$\Longleftarrow$ $\text{visit}\left(v\right)$ נקרא מתוך שרשרת קריאות רקורסיביות מתוך $\text{visit}\left(u\right)$.

$\Longleftarrow$ $\text{visit}\left(v\right)$ מסתיים לפני $\text{visit}\left(u\right)$.

$\Longleftarrow$ $f\left(v\right)<f\left(u\right)$

$\Longleftarrow$ $\boxed{s\left(u\right)<s\left(v\right)<f\left(v\right)<f\left(u\right)}$.

מדוע v הוא צאצא של u?

נוכיח באינדוקציה לפי מספר הקריאות של visit שבוצעו בין $\text{visit}\left(u\right)$ ל-$\text{visit}\left(v\right)$.

<u>בסיס</u>: $\text{visit}\left(v\right)$ בוצע ישירות מתוך $\text{visit}\left(u\right)$.

לפי הגדרת האלגוריתם, $p\left(v\right)=u$, ולכן v צאצא של u.

<u>צעד</u>: נניח כי $\text{visit}\left(v\right)$ נקרא מתוך $\text{visit}\left(w\right)$.

$\Longleftarrow$ $p\left(v\right)=w$, כלומר v הוא ילד ישיר (צאצא) של w.

לפי הנחת האינדוקציה, w הוא צאצא של u, ולכן v צאצא של u.

- <u>מקרה שני</u>: $\boxed{f\left(u\right)<s\left(v\right)}$

נרצה להראות שאנחנו במקרה א'.

חייב להתקיים:

$$s\left(u\right)<f\left(u\right)<s\left(v\right)<f\left(v\right)$$

מכיוון שלא ניתן לסיים צומת לפני שמגלים אותו.

נראה ש-v אינו צאצא של u (המקרה ההפוך - סימטרי):

אם נניח בשלילה ש-v הוא כן צאצא של u, אז צריך להתקיים ש-$\text{visit}\left(v\right)$
מתרחש בשרשרת קריאות רקורסיביות שמקורן ב-$\text{visit}\left(u\right)$, ובפרט $\text{visit}\left(v\right)$ מסתיים לפני סיום $\text{visit}\left(u\right)$, ז"א $f\left(u\right) > f\left(v\right)$
- בסתירה!

∎

מסקנה 1.2 (מטענת העזר)

v צאצא של u ביער ה-DFS $\iff$ $s\left(u\right) < s\left(v\right) < f\left(v\right) < f\left(u\right)$.

משפט 1.3 (אפיון ליחסי אב-צאצא ביער ה-DFS) לכל גרף מכוון G ולכל ריצת DFS,
v צאצא של u ביער ה-DFS, **אם ורק אם** בזמן גילוי u, יש ב-G מסלול מ-u ל-v שכל הצמתים בו הן unvisited (פרט ל-u עצמו).

<u>הוכחה.</u>

<u>$\Leftarrow$</u>: נתון ש-v צאצא של u. נרצה להוכיח שברגע גילוי u
יש ב-G מסלול מ-u ל-v שמכיל רק צמתים שהם unvisited.

יהי P המסלול מ-u ל-v ביער ה-DFS (G_p).
נראה שברגע גילוי u, כל הצמתים ב-P הם unvisited.

יהי w צומת ב-P, לכן w צאצא של u ביער ה-DFS.
לפי המסקנה $s\left(u\right) < s\left(w\right)$,
ולכן ברגע גילוי u כל הצמתים ב-P הם unvisited.

<u>$\Rightarrow$</u>: נתון שברגע גילוי u, קיים ב-G מסלול P מ-u ל-v
שכל הצמתים בו הם unvisited (באותו הרגע). נרצה להראות ש-v צאצא של u.

נניח בשלילה ש-v אינו צאצא של u, ויהי x הצומת הראשון במסלול שאינו צאצא של u (הערה: קיום x מובטח בגלל v, אחרת v צאצא של u).

יהי y הצומת הקודם ל-x במסלול (קיום y מובטח כי x בהכרח אינו הצומת הראשון ב-P). מתקיים:

$$s\left(u\right) \underbrace{<}_{\substack{\text{ברגע גילוי } u\text{, כל הצמתים} \\ \text{ב-}P\text{ הן unvisited.}}} s\left(x\right) \underbrace{<}_{\substack{\text{יש קשת מ-}y\text{ ל-}x\text{,} \\ \text{ולכן לא נסוגים מ-}y \\ \text{עד ש-}x\text{ מתגלה.}}} f\left(y\right) \underbrace{\leq}_{\substack{y\text{ צאצא של }u\text{ (אולי }y=u\text{)} \\ \text{+ מהמסקנה.}}} f\left(u\right)$$

אבל לפי המסקנה x צאצא של u (שכן אינטרוולים לא יכולים להיחתך),
וזו סתירה להנחת השלילה.

∎

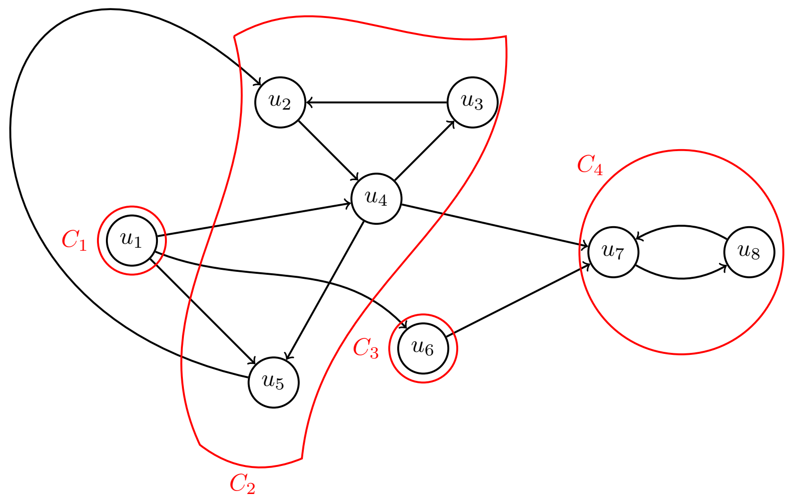
2.6. רכיבים קשירים היטב.

הגדרה 1.9 (רכיב קשיר היטב) נגדיר יחס (רלציה) על זוגות של צמתים באופן הבא:
u ו-v ביחס $\iff$

- ב- G יש מסלול מ-u ל-v.

- ב- G יש מסלול מ-v ל-u.

הרכיבים הקשירים היטב הם מחלקות השקילות של היחס הזה.

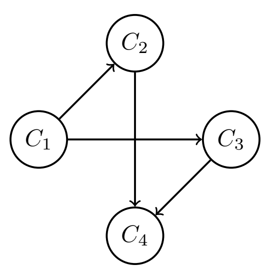


איור 4: רכיבים קשירים היטב עבור גרף לדוגמה

הגדרה 1.10 (גרף רכיבים קשירים היטב) לכל גרף מכוון G ניתן להגדיר את גרף הרכיבים הקשירים היטב שלו $\bar{G}\left(\bar{V},\bar{E}\right)$ להיות:

$$\bar{V}=\left\{C \,\middle|\, C \text{ רכיב קשיר היטב של } G\right\}$$

$$\bar{E}=\left\{(C_i\to C_j)\,\middle|\,\begin{array}{c}\text{קיימים } u\in C_i \text{ ו- } v\in C_j\\ \text{כך ש-}(u_i\to u_j)\in E\end{array}\right\}$$



איור 5: גרף הרכיבים הקשירים היטב של הגרף מהאיור הקודם

דוגמה 1.1 *איור של גרף הרכיבים הקשירים היטב של הדוגמה הקודמת*

הערה 1.4 גרף רכיבים קשירים היטב הוא בהכרח חסר מעגלים מכוונים (גרף א-ציקלי),
ולכן ניתן לבצע עליו מיון טופולוגי.
באופן כללי, נוח לפתור בעיות על גרפים מסוג זה.

שאלה 1.4 (השאלה החישובית שתעניין אותנו) בהינתן גרף מכוון $G=(V,E)$,
כיצד נחשב את גרף הרכיבים הקשירים היטב שלו?

הערה 1.5 קל לפתור את הבעיה בזמן ריבועי, ע"י הרצת אלגוריתם סריקה (BFS, DFS) מכל צומת.
נרצה לפתור את הבעיה בזמן לינארי, בהתבסס על התכונות שמצאנו מקודם.

הערה 1.6 באופן כללי, מובטח שכל קשת אחורית "סוגרת מעגל".

נרצה לבחור נציג לכל רכיב קשיר היטב, שהוא:

- "קנוני".
- "הכי קדמון" $\Longleftarrow$ בעל זמן הנסיגה הגדול ביותר.

הגדרה 1.11 (הנציג של צומת u) בהינתן ריצת DFS נתונה, הנציג של צומת u זה הצומת v שישיג מ-u ב-G, בעל זמן הנסיגה $f(v)$ הגדול ביותר.
<u>מסמנים</u> $\varphi(u)$.

הערה 1.7 כל רכיב קשירות היטב מוכל בהכרח בעץ יחיד ביער ה-DFS (לפי המסקנה ממקודם),
אבל ההפך אינו בהכרח נכון.

למה 1.4 לכל ריצת DFS ולכל צומת u, מתקיים ש-u ו-$\varphi(u)$ באותו רכיב קשיר היטב.

<u>הוכחה</u>. ב-G יש מסלול מ-u ל-$\varphi(u)$ (מהגדרת נציג).
נשאר להראות שב-G יש מסלול מ-$\varphi(u)$ ל-u, ביחס לריצת DFS נתונה.

נראה קיום של מסלול שכזה על ידי כך שנוכיח ש-u הוא צאצא של $\varphi(u)$ ביער ה-DFS
(זוהי טענה חזקה יותר).

אם $u=\varphi(u)$ אז סיימנו, לכן נניח $u\neq\varphi(u)$.

מהנחה זו נובע כי $f(u)<f(\varphi(u))$.
לכן, בזמן גילוי u ע"י ה-DFS, לא יתכן שכבר נסוגנו מ-$\varphi(u)$.
לכאורה, יתכנו 2 אפשרויות:

(1) ברגע גילוי u, $\varphi(u)$ חדש (unvisited).
(2) ברגע גילוי u, $\varphi(u)$ אינו חדש, אבל עדיין לא נסוגנו מ-$\varphi(u)$.

נוכיח ש-(1) אינו אפשרי.

נניח בשלילה ש-(1) אפשרי, ויהי P המסלול לפי ההגדרה ש-$\varphi(u)$ נציג של u.

ברגע גילוי u לא יתכן שכל הצמתים ב-P חדשים
(אחרת, לפי משפט, $\varphi(u)$ צאצא של u, ולכן $f(\varphi(u))<f(u)$, בסתירה להגדרת הנציג).

יהי v הצומת האחרון במסלול P שאינו חדש (visited) ברגע גילוי u ע"י DFS.

לכן, ברגע גילוי v, הסיפא של P מ-v ל-$\varphi\left(u\right)$ כולה חדשה (unvisited).

לכן, $\varphi\left(u\right)$ צאצא של v ביער ה-DFS, ואז:

$$f\left(\varphi\left(u\right)\right)<f\left(v\right)$$

וזו סתירה לכך ש-$\varphi\left(u\right)$ הוא הנציג של u.

לכן לפי משפט האינטרוולים, האינטגרל של u מוכל בזה של $\varphi\left(u\right)$, בפרט u צאצא שלו.

■

טענה 1.2 לכל גרף מכוון $G=(V,E)$ ולכל שני צמתים $u,v\in V$ ולכל ריצת DFS, מתקיים:

u,v באותו רכיב קשיר היטב $\iff \varphi(u)=\varphi(v)$

הוכחה.

$\Leftarrow$: אוסף הצמתים שישיגים מ-u זהה לאוסף הצמתים שישיגים מ-v,
ולכן בהכרח יש ל-u,v אותו נציג.

$\Rightarrow$: לפי טענת העזר, $u,\varphi(u)$ באותו רכיב קשיר היטב.
באופן דומה, $v,\varphi(v)$ באותו רכיב קשיר היטב.
אבל $\varphi(u)=\varphi(v)$ מהנתון, ולכן u,v באותו רכיב קשיר היטב. ■

2.7. האלגוריתם למציאת רכיבים קשירים היטב.

(1) מריצים DFS על $G=(V,E)$ לקבלת זמני נסיגה $f(u)$ לכל צומת $u\in V$.
(2) נסמן ב-G^R את הגרף שמתקבל מ-G ע"י הפיכת כיווני הקשתות.
(3) מריצים DFS על G^R. כאשר מתחילים עץ חדש ביער ה-DFS, בוחרים את הצומת שנותר עם זמן הנסיגה הגדול ביותר משלב 1 של האלגוריתם.

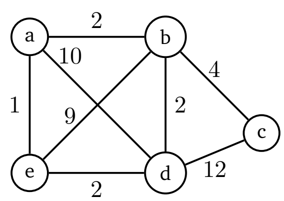
- הקלט: גרף $G=(V,E)$.
- הפלט: העצים שמתקבלים בריצת ה-DFS השנייה על G^R (שלב 5).
  הם הרכיבים הקשירים היטב של G.

פרק 2

עצים פורשים מינימליים

1. בעיות אופטימיזציה ברשתות

דוגמה 2.1 נתונה רשת התקשורת הבאה:



איור 1: על כל קשת מופיע מחיר השימוש בה.

נניח כי הצומת a מעוניין להפיץ הודעה לכל הצמתים ברשת.
יש למצוא תת קבוצת של קשתות ברשת, שעליהן ההודעה תעבור כך שתגיע לכל הצמתים.

שאלה 2.1 האם יכול להיות שבתת-הגרף שנבחר יהיו מעגלים?

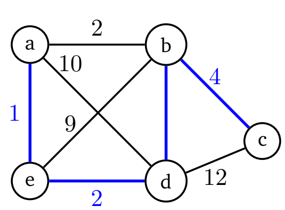
הערה 2.1 נשים לב כי היות שנרצה להשיג מחיר מינימלי, תת-הגרף שהתקבל מבחירת הקשתות הינו לבטח חסר מעגלים.

2. בעיית עץ פורש מינימום (עפ"מ)

הגדרה 2.1 (בעיית עץ פורש מינימלי) נתון גרף קשיר לא מכוון $G=(V,E)$, שבו לכל קשת (v,u) יש משקל $w\left(v,u\right)$.

יש למצוא עץ פורש של הגרף, שסך משקל הקשתות שלו מינימלי.

דוגמה 2.2 (דוגמה לעץ פורש של גרף משקלים נתון)



איור 2: עץ פורש של הדוגמה הנתונה.

- (a,e) הקשת הזולה ביותר שיוצאת מ-a, לכן נסמנה בכחול.
- נסמן את הקשת (e,d).
- נסמן את הקשת (b,d).
- נסמן את הקשת (b,d).
- נסמן את הקשת (b,c).

כאשר משקל העץ הפורש הינו 9.

ללא הוכחה, נציין שזהו גם למעשה עץ פורש מינימלי.

נבחין שזהו אינו העץ הפורש היחיד בעל משקל 9, שכן היה ניתן למשל להחליף את הקשת (e,d) בקשת (a,b) ולקבל תוצאה דומה.

3. אלגוריתמים לבעיית עץ פורש מינימום

נראה אלגוריתם גנרי, ובהמשך נציג אלגוריתמים שמתקבלים כמקרים פרטיים של אלגוריתם זה.

- הרעיון: נשתמש באלגוריתם חמדן שיבנה עפ"מ קשת אחר קשת, ע"י הוספת קשתות עם משקל נמוך והשמטת קשתות עם משקל גבוה.
- האלגוריתם יתקדם ע"י צביעת קשתות: קשתות שיצבעו בכחול יופיעו בעץ, וקשתות שיצבעו באדום יושמטו.
- האלגוריתם יקיים בכל שלב את שמורת הצבע.

טענה 2.1 (שמורת הצבע) קיים עפ"מ שמכיל את כל הקשתות הכחולות ואף אחת מהקשתות האדומות.

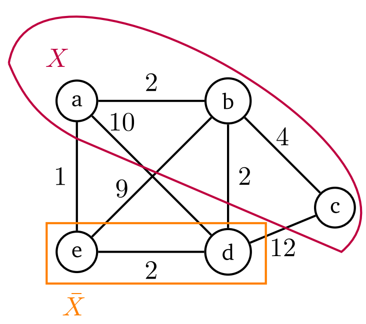
מסקנה 2.1 משמורת הצבע נובע כי כאשר כל הקשתות ב-G נצבעו, הקשתות הכחולות יוצרות עץ עפ"מ.

הגדרה 2.2 (חתך/cut) בגרף $G=(V,E)$ הוא חלוקה של קבוצת הצמתים V לשתי תתי קבוצות: X ו-$\bar{X}=V\setminus X$.

הגדרה 2.3 (חצייה של קשת את חתך הגרף) נאמר שקשת חוצה את החתך אם קצה אחד שלה ב-X והקצה האחר ב-$\bar{X}$.
לפעמים נגיד שקשת כזו תהיה קשת של החתך.

דוגמה 2.3 (דוגמה לחתך ולקשתות שחוצות אותו) נגדיר חתך ברשת: $X=\{a,b,c\}$
הקשתות שחוצות את החתך:

$\{(b,d),(a,e),(c,d),(a,d),(b,e)\}$



איור 3: החתך לדוגמה על גרף הרשת.

3.1. האלגוריתם הגנרי למציאת עפ"מ.

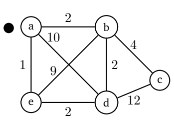
הגדרה 2.4 (הכלל הכחול) יהי $X \subseteq V$ כך שאין קשת כחולה שחוצה את $(X, \bar{X})$.
אזי ניתן לצבוע בכחול את הקשת הקלה ביותר שאינה צבועה מבין אלו שחוצות את $(X, \bar{X})$.

הגדרה 2.5 (הכלל האדום) יהי C מעגל שאין בו קשת אדומה.
אזי ניתן לצבוע באדום את הקשת הכבדה ביותר שאינה צבועה מבין קשתות המעגל C.

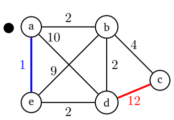
<u>האלגוריתם הגנרי:</u>

- אתחל את כל הקשתות ב-E ללא צבועות.
- כל עוד יש ב-E קשת שאינה צבועה, הפעל את הכלל הכחול או האדום לצביעת אחת מהקשתות.
- הקשתות הכחולות הן עפ"מ.

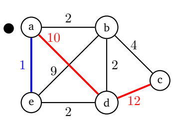
דוגמה 2.4 (דוגמת הרצה)
נחזור לדוגמת הרשת, ונבצע דוגמת הרצה של האלגוריתם החמדן:



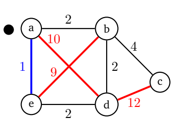
- הכלל הכחול: עם $X = \{a\}$.
נבחר את הקשת הקלה ביותר שאינה צבועה היוצאת מ-a ונצבע אותה בכחול: זוהי (a, e).



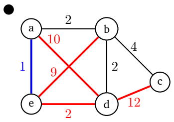
- הכלל האדום על $\{b, c, d\}$:
נבצע באדום את הקשת הכבדה ביותר במעגל חסר הקשתות האדומות $b \to c \to d$, זוהי (d, c).



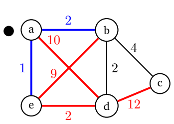
- הכלל האדום על $\{a, b, d\}$:
נבצע באדום את הקשת הכבדה ביותר במעגל חסר הקשתות האדומות $a \to b \to d$, זוהי (a, d).



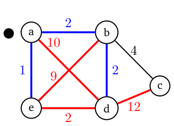
- הכלל האדום על $\{e, b, d\}$:
נבצע באדום את הקשת הכבדה ביותר במעגל חסר הקשתות האדומות $e \to b \to d$, זוהי (e, b).



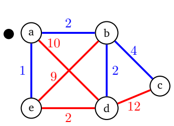
- הכלל האדום על $\{a, b, d, e\}$:
נבצע באדום את הקשת הכבדה ביותר במעגל חסר הקשתות האדומות $a \to b \to d \to e$.
ניתן לבחור כל אחת מהקשתות בעלות משקל 2, למשל (d, e).



- הכלל הכחול: עם $X = \{a, e\}$.
נבחר את הקשת הקלה ביותר שאינה צבועה היוצאת מ-$\{a, e\}$ ונצבע אותה בכחול: זוהי (a, b).



- הכלל הכחול: עם $X = \{c, d\}$.
נבחר את הקשת הקלה ביותר שאינה צבועה היוצאת מ-$\{c, d\}$ ונצבע אותה בכחול: זוהי (b, d).



- הכלל הכחול: עם $X = \{c\}$.
נבחר את הקשת הקלה ביותר שאינה צבועה היוצאת מ-$\{c\}$ ונצבע אותה בכחול: זוהי (b, c).

ואכן, קיבלנו כי משקל העפ"מ הינו 9.

הערה 2.2 (ללא הוכחה) בכל שלב באלגוריתם הגנרי, הקשתות הכחולות יוצרות <u>יער</u> של עצים כחולים, שכן הקשתות הכחולות תמיד מוכלות באיזשהו עפ"מ של הגרף.

הגדרה 2.6 (אי שייכות צומת לעץ כחול) נאמר שצומת v אינו בעץ כחול, אם לא קיימת קשת שנוגעת בצומת v שצבועה בכחול.

3.1.1. נכונות האלגוריתם הגנרי.

שאלה 2.2 האם האלגוריתם תמיד מצליח לצבוע את כל הקשתות?

שאלה 2.3 האם מובטח שבסיום האלגוריתם הקשתות הכחולות יגדירו עפ"מ?

למה 2.1 (הבחנה על עצים פורשים) יהי T עץ פורש של G.
אם נוסיף ל-T קשת $e \notin T$, נקבל ב-T מעגל יחיד C.
אם נשמיט מ-C קשת, בוודאות נקבל שוב עץ פורש של G.

משפט 2.1 (נכונות האלגוריתם הגנרי) קיים עפ"מ T שמכיל את כל הקשתות הכחולות ואף אחת מהקשתות האדומות.

משפט 2.2 (כל הקשתות נצבעות + נכונות שמורת הצבע)
האלגוריתם הגנרי צובע את כל הקשתות של **גרף קשיר** G, ומקיים את "שמורת הצבע".

<u>הוכחת המשפט.</u>

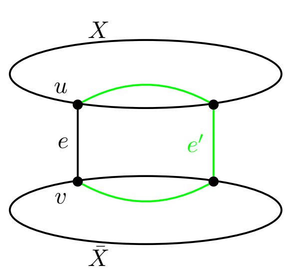
(1) הראינו כי האלגוריתם מקיים את שמורת הצבע אחרי הפעלה של הכלל הכחול.

■

<u>הוכחה.</u> נראה תחילה כי האלגוריתם מקיים את השמורה, באינדוקציה על מספר האיטרציות (הפעלות של הכלל האדום או הכחול):
<u>בסיס האינדוקציה:</u> בתחילה אף קשת לא צבועה, ולכן כל עפ"מ ב-G מקיים את השמורה (כלומר, הטענה נכונה באופן ריק).

<u>צעד האינדוקציה:</u> נטפל לחוד בשני מקרים:

(1) נניח כי השמורה מתקיימת לפני הפעלה של הכלל הכחול.
תהי e קשת שנצבעת כעת בכחול, ויהי T עפ"מ שקיים את השמורה לפני שהקשת e נצבעה.
אם $e \in T$, אזי T מקיים את השמורה אחרי שהקשת e נצבעה (סיימנו).
אם $e \notin T$, נסתכל על החתך $X, \bar{X}$ שעליו הפעלנו את הכלל הכחול.



יש מסלול בעץ T שמחבר בין הצמתים u,v בקצוות של הקשת e.
היות ש-e חוצה את החתך, קיימת על המסלול הנ"ל קשת אחרת e' שחוצה את החתך.

מהנחת האינדוקציה אין ב-T קשת אדומה, שכן הוא מקיים את שמורת הצבע.
מהכלל הכחול (בחרנו חתך ללא קשתות חוצות כחולות), נקבל גם כי e' לא צבועה בכחול.
לכן, e' אינה צבועה.

בנוסף, בהכרח $w\left(e'\right)\geq w\left(e\right)$ (שכן e נבחרה להיות הקשת החוצה בעלת משקל מינימלי. למעשה, בהכרח יש שוויון).
לכן, נוכל להשמיט את הקשת e' מהעץ T ולהוסיף במקומה את e.

נשים לב כי אם המסלול היחיד בין שני צמתים ב-T עבר קודם דרך e', המסלול יעבור כעת דרך e. בנוסף, T נשאר עפ"מ, כי המשקל הכולל של הקשתות בעץ לא עלה.

אם נצבע כעת את e בכחול, נקבל כי השמורה מתקיימת עבור T החדש.

(2) נניח כי השמורה מתקיימת לפני הפעלה של הכלל האדום.
תהי E קשת שנצבעת כעת באדום, ויהי T עפ"מ שמקיים את השמורה לפני שהקשת e נצבעת.

אם $e\notin T$, אזי T מקיים את השמורה גם אחרי שהקשת e נצבעת.

נניח ש-$e\in T$. אזי, השמטת e מ-T מחלקת את T לשני עצים ומגדירה חלוקה של הצמתים ב-G.

איור
המעגל שעליו הפעלנו את הכלל האדום מכיל מסלול נוסף בגרף G מ-u ל-v.

על המעגל יש קשת $e'=\left(x,y\right)$, שחוצה את החתך $\left(T_1,T_2\right)$.

מהשמורה נובע ש-e' אינה כחולה כי $e'\notin T$, ומהכלל האדום נובע כי e' גם אינה אדומה.

בנוסף, מהכלל האדום נובע $w\left(e'\right)\leq w\left(e\right)$.

הוספת e' ל-T והשמטת e יוצרת עץ פורש חדש (מבחנה 1, אם נוסיף... אם נשמיט...□).
בנוסף, לא הגדלנו את משקל העץ. לכן T החדש עפ"מ.

נראה כעת כי האלגוריתם צובע את <u>כל</u> הקשתות ב-G:
נניח בשלילה שיש קשת e לא צבועה, אבל אי אפשר להפעיל אף אחד מהכללים.
לפי הכלל הכחול, הקשתות הכחולות יוצרות <u>יער</u> של עצים כחולים, שכן הקשתות הכחולות תמיד מוכלות באיזשהו עפ"מ של הגרף.
נבחין בין 3 מקרים לגבי $e=\left(u,v\right)$:

(1) אם שני הקצוות של e באותו עץ כחול, אזי נקבל: *איור*
מצאנו בגרף G מעגל בלי קשתות אדומות, שמכיל את e ואת המסלול בעץ הכחול בין u ל-v.
לכן ניתן להפעיל את הכלל האדום.

(2) אם הקצוות של e בעצם כחולים שונים: *איור*
נסמן ב-X את קבוצת הצמתים ב-T_1, וב-$V \setminus X$ את שאר הצמתים.
קיבלנו חתך ב-G שאין בו קשתות כחולות, לכן נוכל להפעיל את הכלל הכחול.

(3) לפחות קצה אחד של e אינו בעץ כחול, בה"כ נניח כי $e = (u, v)$ ו-v אינו בעץ כחול.
נגדיר $X = \{v\}$ ו-$\bar{X} = V \setminus X$.
מצאנו חתך ללא קשת כחולה, ולכן ניתן להפעיל את הכלל הכחול.

$\Longleftarrow$ כל עוד יש ב-G קשת לא צבועה, מובטח שנוכל להפעיל את אחד הכללים, ולכן האלגוריתם צובע את כל הקשתות.

■

3.2. האלגוריתם של Prim. נתון גרף קשיר לא מכוון $G = (V, E)$.

(1) <u>אתחול:</u> כל הקשתות לא צבועות, נבחר $T = \{r\}$ כאשר r צומת כלשהי.
(2) כל עוד $T \neq V$, בצע:
- תהי $E = (u, v)$ הקשת הקלה ביותר שחוצה את החתך $(T, V \setminus T)$, כך ש-$u \in T$.
- צבע את e בכחול ובצע $T := T \cup \{v\}$.

משפט 2.3 (נכונות אלגוריתם Prim) T הוא עפ"מ ב-G.

<u>הוכחה.</u> נראה כי האלגוריתם של Prim הוא מימוש של האלגוריתם הגנרי.
נתייחס לכל קשת שלא נוספה ל-T בסיום האלגוריתם כקשת שנצבעת באדום.

נסתכל על קשת $e = (u, v)$ שהאלגוריתם מוסיף ל-T באיטרציה כלשהי.
באיטרציה זו, אין קצת כחולה שחוצה את החתך $(T, V \setminus T)$.
בנוסף, הקשת e חוצה את החתך, והיא הקלה ביותר שחוצה את החתך הנ"ל ("בין אלו שאינן צבועות").
לכן צביעת e היא חוקית לפי הכלל הכחול.

נבחין את הקשתות שאינן ב-T בסיום האלגוריתם.
כל קשת e <u>שלא נוספה ל-T</u> סוגרת מעגל ב-T. במעגל זה הינה הקשת היחידה שאינה צבועה, ושאר הקשתות בהכרח כחולות.
לכן צביעת e באדום היא הפעלה חוקית של הכלל האדום.

סה"כ קיבלנו שהאלגוריתם של Prim הוא מימוש ספציפי של האלגוריתם הגנרי.

■

3.2.1. סיבוכיות אלגוריתם Prim.

שאלה 2.4 מהי סיבוכיות האלגוריתם?

המפתח לשימוש יעיל של אלגוריתם Prim הוא לבחור בקלות קשת בעלת משקל מינימלי מבין אלו שחוצות את החתך.
נחזיק את כל הצמתים שאינן ב-T בתור עדיפויות Q.

לכל צומת v מוחזק מפתח $\text{key}\left(v\right)$ - המשקל המינימלי של איזושהי קשת שמחברת את v ל-T.
אם אין קשת כזאת, אז מסמנים $\text{key}\left(v\right)=\infty$. בסיום האלגוריתם התור ריק.

אם $v\in Q$ ו-$\text{key}\left(v\right)\neq\infty$, אז $\left(v,\underbrace{\pi\left(v\right)}_{\substack{\pi\left(v\right)\text{ היא}\\ \text{צומת ב-}T.}}\right)$ היא הקשת הקלה בין v ל-T.

נממש את Q בעזרת ערימת מינימום (heap).

שאלה 2.5 מה הפעולות שנבצע על הערימה?

(1) <u>אתחול</u>: לכל צומת $v\in G$ מגדירים: $\begin{matrix}\text{key}\left(v\right)\leftarrow\infty\\ \pi\left(v\right)\leftarrow\text{NULL}\end{matrix}$
(2) מצא בערימה את המפתח המינימלי, נניח כי הינו שייך לצומת $u\in V\setminus T$.
(3) הוסף את u ל-T.
(4) לכל שכן v של u כך ש-$v\notin T$,
אם $\text{key}\left(v\right)>w\left(u,v\right)$, בצע פעולת Decrease Key, ועדכן $\pi\left(v\right)\leftarrow u$.

<u>סיבוכיות כל אחד מהשלבים:</u>

- צעד (1): מתבצע ב-$O\left(\left|V\right|\right)$.
- הוצאת המפתח המינימלי בצעד (2), ב-$O\left(\log\left|V\right|\right)$.
- תתבצע $\left|V\right|$ פעמים, סה"כ $O\left(\left|V\right|\log\left|V\right|\right)$.
- פעולת Decrease Key בצעד (4) תתבצע לכל היותר עבור כל השכנים של כל צומת, לכל היותר $\left|E\right|$ פעמים.

<u>סיבוכיות האלגוריתם:</u>

$$O\left(\left|V\right|\log\left|V\right|\right)+O\left(\left|E\right|\log\left|V\right|\right)=O\left(\left|E\right|\log\left|V\right|\right)$$

3.3. האלגוריתם של Kruskal.

(1) כל הקשתות לא צבועות, מיין את הקשתות בסדר לא יורד לפי משקלן, $F=\emptyset$.
(2) תהי e הקשת הבאה לפי סדר המיון.
אם e סוגרת מעגל בעץ כחול, צבע אותה באדום.
אחרת צבע את e בכחול, ובצע הוספה של e ל-F.
(3) החזר כפלט את אוסף הקשתות ב-F.

משפט 2.4 (נכונות) הגרף $\left(V,F\right)$ שמורכב מכל הצמתים ב-G ומהקשתות F שאלגוריתם Kruskal החזיר, הוא עפ"מ ל-G.

הוכחה. נראה כי Kruskal מבצע הפעלה חוקית של הכלל הכחול או האדום.

אם e סוגרת מעגל בעץ כחול, אז מצאנו מעגל שאין בו קשתות אדומות.
היות ש-e היא היחידה שאינה צבועה, היא גם המקסימלית בין הקשתות שאינן צבועות על המעגל.
נפעיל את הכלל האדום.

אם $e=(u,v)$ לא סוגרת מעגל ב-F, אז נבחין בין שני מקרים:

(1) קצה אחד של e אינו בעץ כחול, בה"כ נניח כי זהו הצומת v.
נגדיר $X=\{v\}$ ואת $\bar{X}=V\setminus X$.
מצאנו חתך שלא חוצות אותו קשתות כחולות. נפעיל את הכלל הכחול.
מבין הקשתות הלא צבועות שחוצות את $\left(X,\bar{X}\right)$, היא בעלת משקל מינימלי (בגלל המיון).
איור

(2) בשני הקצוות של e יש עצים כחולים.
ניקח עץ מאחד הקצוות, נסמנו ב-T_1, ונגדיר $X=T_1$, ואץ $\bar{X}=V\setminus X$.
איור
קיבלנו חתך שלא חוצות אותו קשתות כחולות.
הקשת e בעלת משקל מינימלי בין הקשתות שחוצות את החתך $\left(X,\bar{X}\right)$ ואינן צבועות, עקב המיון. נפעיל את הכלל הכחול.

$\boxed{\Longleftarrow \text{ Kruskal הוא מימוש ספציפי של האלגוריתם הגנרי.}}$ ■

סיבוכיות האלגוריתם: ניתן לממש את Kruskal ב-$O\left(|E|\log|V|\right)$.