אלגוריתמים 1: הרצאה 2 ־ MST

5 בנובמבר 2025

גיא יער־און.

0.1 עץ פורש מינימום

עץ פורש מינימום, או MST (Minimum spanning tree) הוא מה שנעסוק בו בהרצאה זו.

למה צריך גרפים? למשל, עבור רשתות תקשורת. כל קודקוד בעץ הוא מחשב ברשת, וכל קשת בין הקודקודים מציינת האם יש תקשורת ישירה בין שני המחשבים ברשת. אנחנו חברה, שרוצה למכור רשתות תקשורת, והמטרה שלנו היא שהקשתות שנבחור ישאירו את הגרף קשיר. כלומר ־ נרצה לבחור קשתות כרצוננו רק נשים לב שבכל שלב נתון נוכל להגיע לכל מחשב ברשת. נשים לב ־ שבמהלך הבחירה שלנו יתכן ונבחור קשתות מיותרות, בשביל חלילה לא להגיע למצב שהגרף לא קשיר.

עץ פורש: תת גרף של הגרף המקורי, שהוא קשיר וללא מעגלים (עץ). כלומר $V' = V$ וכן $E' \subseteq E$.

ייצוג לגרפים: מטריצת שכנויות, רשימה ($List$) של שכנויות.

נשים לב כי נרצה למצוא את העץ הפורש הנ"ל שיאפשר לנו למצוא דרך להגיע לכל המחשבים (קודקודים) ברשת (גרף). נשים לב ־ כי ייתכן ויהיה כמה דרכים לעשות זאת, כמו כן ־ ייתכן ולכל מעבר ברשת מסוימת יהיה מחיר שונה (כלומר, לכל קשת יהיה מחיר שנשלם שנעלה עליה). ולכן כל קשת תסומן אצלנו בערך מספרי מסויים.

בהינתן x קודקודים בעץ הפורש, נרצה למצוא $x-1$ קשתות שהעלות שלהן יחד היא הנמוכה ביותר.

נשים לב כי כאשר נעבוד עם עצים פורשים מינימום, נניח מראש כי הגרף יהיה קשיר.

יהי $G=(V,E)$ מולטי גרף קשיר ממושקל עם פונקציית משקל על הקשתות $w: E \to \mathbb{R}$.
יהי $T=(V,E_T)$ עץ פורש של G. אזי, נאמר שהמשקל של T הוא

$$w(T) = \sum_{e \in E_T} w(e)$$

עץ פורש מינימום של G הינו עץ פורש שמשקלו הוא הקטן ביותר.

מסקנה: יהי T עפ"מ, אזי לכל עץ פורש $T' \in G$ מתקיים $w(T') \geq w(T)$.

הערה: מולטי גרף הוא גרף בו קבוצת הקשתות הינה *multi-set* , כלומר בין שני קודקודים

בגרף, יכולות לעבור מספר קשתות. ומדוע שנרצה להשתמש בו? הרי ברור כי כאשר נחפש עפ"מ, בהינתן שני קודקודים v, u וקשתות e_1, e_2 שמשקלן בהתאמה $1, 2$, נרצה לבחור בקשת שמשקלה 1. בהמשך, נבין מדוע האלגוריתם פועל על מולטי גרף למרות שזה לא נראה אינטואטיבי.

הערה2: גרף לא מכוון הוא גרף בו התנועה דו כיוונית, אם קיימת $e: v \to u$ בהכרח אפשר לנוע גם ב־$u \to v$. גרף מכוון הוא זה בו התנועה מוגדרת בכיוון מסויים. כלומר זה שקיימת $e: v \to u$ לא גורר שניתן ללכת בכיוון $u \to v$.

0.2 בעיית מציאת עץ פורש מינימום

קלט: גרף לא מכוון קשיר $G = (V, E)$ ופונקציית משקל $w : E \to \mathbb{R}$

פלט: עץ פורש מינימום של G (ביחס לפונקציית משקל w).

0.3 אלגוריתים חמדניים $(Greedy)$

מקבלים החלטה לוקאלית וממשיכים רקורסיבית בלי לשנות את ההחלטה ובלי לדעת מה הפתרון ברקורסיה. למשל - חיפוש בינארי.

למת הבחירה החמדנית: הבחירה שהאלגוריתם ביצע באופן חמדני לא מונעת ממנו להגיע לפתרון האופטימלי.

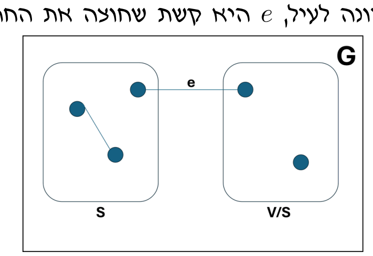
למת תת המבנה האופטימלי: מכלול בחירות חמדניות יביא את התוצאה האופטימלית.

0.4 למת הבחירה החמדנית

חתך: יהי $G = (V, E)$ גרף ויהי $S \subset V$ כך ש־$S \neq \emptyset$. החלוקה $(S, V/S)$ נקראת חתך של G והקשתות $\{e = (u, v) | u \in S, v \in V/S\}$ נקראות **קשתות שחוצות את החתך / קשתות בחתך**.

נשים לב - S לעולם לא תהיה שווה ל־V, ולעולם לא תהיה ריקה.

בתמונה לעיל, e היא קשת שחוצה את החתך.



קשת קלה ביותר בחתך: יהי $G = (V, E)$ גרף לא מכוון עם פונקציית משקל $w : E \to \mathbb{R}$. יהי $(S, V/S)$ חתך של G. קשת $(u, v) \in E$ נקראת קשת קלה ביותר בחתך $(S, V/S)$ אם לכל קשת e' שחוצה את החתך מתקיים $w(e) \le w(e')$. (נשים לב, יתכן שיש כמה קשתות כאלו באותו משקל שהן הקלות ביותר).

למה 1: יהי $G = (V, E)$ מולטי גרף קשיר עם פונקציית משקל $w : E \to \mathbb{R}$. לכל $\emptyset \neq S \subset V$, לכל קשת e קלה בחתך $(S, V/S)$ קיים עפ"מ שמכיל את e.

הוכחה: יהי $G = (V, E)$ מולטי גרף קשיר עם פונקציית משקל $w : E \to \mathbb{R}$.

יהי $(S, V/S)$ חתך של G, ותהי $e = (u, v)$ קשת קלה בחתך, כאשר $u \in S, v \in V/S$.

יהי T עפ"מ של G. (בהכרח קיים כזה, כיוון שG קשיר, לכן יש לו עצים פורשים, ובהכרח אחד מהם מינימלי). אם $e \in T$ אזי סיימנו.

אחרת, $e \notin T$. לפי תכונות של עצים ־ קיים מסלול פשוט P (כל הקודקודים בו, ופעם אחת בלבד) יחיד בT מu אל v. מסלול פשוט זה, לא מכיל את e כיוון ש$e \notin T$.

בP חייבת להיות קשת שחוצה את החתך. אחרת, כל קשת שנעבור בה תשאיר אותנו בחתך, ואז לא נוכל להגיע אל הצד השני של העץ, מעבר לחתך, שבהכרח יש שם קודקודים כיוון ש$\emptyset \neq S \subset V$.

נסמן ב$e' = (u', v')$ את הקשת הראשונה בP שחוצה את החתך. כלומר, $P : (P_1^{u \rightsquigarrow u'} \to e' \to P_2^{v' \rightsquigarrow v})$

נבנה $T' = (V, E_{T'})$ באשר $E_{T'} = E_T \cup \{e\} \setminus \{e'\}$. כעת נוכיח כי T' הוא MST.

1. נוכיח T' הוא עץ פורש: עלינו להוכיח כי הוא קשיר וכן כי $|E_{T'}| = |V| - 1$, ואז בהכרח הוא עץ פורש. נשים לב כי

$$|E_{T'}| = |E_T| + 1 - 1 = |E_T|$$

כיוון שT הוא עפ"מ, הוא בהכרח ע"פ ולכן $|E_T| = |V| - 1$, ומכאן ש$|E_{T'}| = |E_T| = |V| - 1$.

נוכיח קשירות. יהיו $x, y \in V$. נרצה להוכיח כי קיים מסלול בT' בין x לy. T הוא ע"פ, לכן קיים מסלול פשוט יחיד מx לy. נסמן את המסלול בP'.

אם $P' \notin e'$, אזי המסלול P' קיים גם בעץ T' ולכן יש מסלול בין x לy. (כלומר, הצלע שהורדנו לא נמצאת על המסלול בין השניים).

אם $e' \in P'$ (כלומר, הצלע שהורדנו בבניית העץ נמצאת על המסלול הפשוט), נניח בה"כ כי u' מופיע לפני v' בP' ונסמן את המסלול $x \rightsquigarrow u'$ בP_1' ואת המסלול $y \rightsquigarrow v'$ בP_2'. כלומר,

$$P' : (P_1^{x \rightsquigarrow u'} \to e' \to P_2^{v' \rightsquigarrow y})$$

כעת נבנה מסלול בT' מx לy באופן הבא:

$$P'_{1x \to u'} \rightsquigarrow (P_1^R)_{u' \to u} \rightsquigarrow e_{u \to v} \rightsquigarrow (P_2^R)_{v \to v'} \rightsquigarrow (P_2')_{v' \to y}$$

הערה. P_1 הוא המסלול שמוביל אותנו מ$u \to u'$. נרצה לכתוב R (רוורס) כי נרצה ללכת כעת במסלול ההפוך. בדומה עבור P_2^R.

סה"כ, בנינו מסלול מוכל בT' שעובר מx אל y, לכן העץ קשיר, ולכן T' הוא עץ פורש של G.

2. נוכיח כי $w(T') \leq w(T)$, כיוון ש$w(T)$ הוא המינימלי, ואם $w(T') \leq w(T)$ הוא קטן מהמינימלי ובפרט מכולם. נשים לב כי

$$w(T') = w(E_T \cup \{e\} \setminus \{e'\}) = w(T) + w(e) - w(e')$$

נתון כי e קשת קלה ביותר, ולכן $w(e) \leq w(e')$ כלומר $w(e) - w(e') \leq 0$ ולכן

$$w(T') = w(T) + w(e) - w(e') \leq w(T)$$

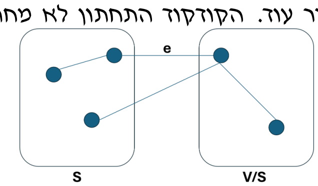
כיוון שהורדנו משהו מ$w(T)$. סה"כ, $w(T') \leq w(T)$, מצד שני T עפ"מ ולכן המשקל שלו קטן משל כל ע"פ אחר, ולכן המשקל של T' הוא הקטן ביותר.
סה"כ T' הוא עץ פורש מינימום, שמכיל את הקשת e.

$\blacksquare$.

0.5 כיווץ קשתות

כרעיון בסיסי מאוד, בהתחשב בלמה שעמלנו קשות להוכיח לפני עמוד, נוכל למצוא את הקשת הקלה ביותר בחתך, לפי הלמה היא נמצאת בMST, ולהפעיל רקורסיה על צד S ורקורסיה על צד V/S וככה באופן רקורסיבי בשיטת הפרד ומשול למצוא את העץ הפורש. **זה לא עובד.** למה?

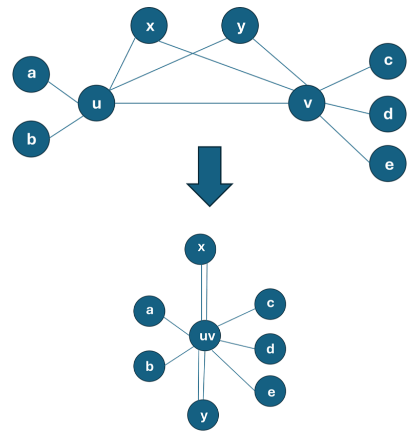
נתבונן בתמונה. מדובר בגרף קשיר. e היא הקשת הקלה ביותר. על פניו ־ אחלה. נעשה ברקורסיה כפי שאמרנו, כעת, כאשר נפעיל ברקורסיה על החתך (צד S), נקבל שהגרף איננו קשיר עוד. הקודקוד התחתון לא מחובר לקודקודים העליונים. ולכן ־ זו אינה תת בעיה.



בעיה נוספת שנצטרך לטפל בה ־ איך נמנע ממצב בו כל הקשתות בגרף בעלות אותו ערך, ו נניח, לפי שיטה בו בוחרים קשת קלה ביותר ומתקדמים ־ במצב זה נתקע כי כל הקשתות בגרף באותו גודל.

כיווץ קשתות:

התהליך יהיה די פשוט: נניח ונרצה להעלים" את הקשת בין $u \to v$, כל שנעשה יהיה כמו בתרשים מטה: נאחד את u, v לקודקוד משותף בשם uv, את השכנים (הלא משותפים שלהם) נחבר באמצעות הוספת קשת לuv בין כל אחד מהשכנים. את השכנים המשותפים שלהם (x, y) נחבר לuv באמצעות הוספת שתי קשתות. כל קשת תקבל את הערך שהיה לה מקודם עם u, v.



פורמלית: יהי מולטי גרף $G=(V,E)$ לא מכוון, פעולת כיווץ על קשת $e=(u,v)$ מייצרת גרף חדש $G/e=(V/e,E/e)$ כך ש:

$$V/e = V \cup \{uv\} \setminus \{u,v\}$$

נגדיר פונקציה: $F:V\to V/e$ כך ש:

$$F(x) := \left\{ \begin{array}{cc} x & x\neq v,u \\ uv & x=u\vee x=v \end{array} \right\}$$

הפונקציה F ממפה את הקודקודים המקוריים, לקודקוד החדש שמייצג אותם.
כעת נגדיר את E/e:

$$E/e := \{(F(x),F(y))|(x,y)\in E\}$$

כך למשל: אם $x=u$ וכן $y\neq u,v$. בהפעלה הבאה של הפונקציה נקבל $(F(x),F(y))=(uv,y)$ כיוון שכל מה שנשלח ל u כעת נמצא ב uv.

נשים לב כי - $(F(u),F(v))=(uv,uv)$ וזו לולאה עצמאית. עם זאת - **בגרף לא מכוון אסור לולאות עצמאיות.** לכן, בתהליך הכיווץ נעלמת קשת אחת - אין לולאה עצמית, הקשת שהייתה בין u ל v איננה עוד. כמו כן, אם נתקלנו בגרף בו יש שלוש קשתות בין u ל v - כולן נעלמות בתהליך הכיווץ, לא רק אחת מהן.

0.6 אלגוריתם גנרי של MST

כעת נדון באלגוריתם גנרי, אין לנו עניין בזמן הריצה שלו ואין לנו דרך להריץ אותו - כשמו כן הוא. חסרים כאן יותר מדי פרטים טכניים וחשובים לזמן הריצה ולהפעלה שלו. אם כן, הוא חשוב כרעיון כללי עליו נתבסס בהמשך. להלן האלגוריתם -

MST(G=(V,E),w):

1. $E_T = \emptyset$

2. while $|V| > 1$:
3. let e be the minimum weight edge of **some** cut in G
4. Add e to E_T
5. Contract e

6. Return E_T

נשים לב ־ שהאלגוריתם ממש מתבסס על למת הבחירה החמדנית, שקיים תמיד עפ"מ אם נבחר את הקשת הקטנה ביותר בחתך כלשהו. מה שנצטרך לעשות ־ זה לדבר בתכונת תת המבנה האופטימלי על סדרה של בחירות.

0.7 תכונת תת המבנה האופטימלי

למה2: יהי $G = (V, E)$ מולטי גרף קשיר, עם פונקציית משקל $w : E \to \mathbb{R}$. יהי $(S, V/S)$ חתך בG. ויהי $e = (u, v)$ קשת קלה בחתך $(S, V/S)$. יהי $T' = (V/e, E_{T'})$ עפ"מ עבור G/e. יהי $E_T = E_{T'} \cup \{e\}$. אזי, $T = (V, E_T)$ הוא עפ"מ עבור G.

(כלומר, קח את הקשת הקלה ביותר e, תכווץ אותה מהגרף ותקבל עפ"מ חדש של G/e. אם תקח בכל פעם את הקשת הזו ותחבר אותה לקבוצת הקשתות הקודמות שיצרו עפ"מ, אתה תקבל עפ"מ עבור G. כלומר ־ זה בדיוק הצעד ברקורסיה שמתבצע שוב ושוב, מה שקורה באלגוריתם שמופיע כאן מעלה, שיוצר לנו עץ פורש מינימום).

הוכחה:

יהי $G = (V, E)$ מולטי גרף קשיר, עם פונקציית משקל $w : E \to \mathbb{R}$. יהי $(S, V/S)$ חתך בG. ויהי $e = (u, v)$ קשת קלה בחתך $(S, V/S)$. יהי $T' = (V/e, E_{T'})$ עפ"מ עבור G/e. יהי $E_T = E_{T'} \cup \{e\}$.

נרצה להוכיח כי T הוא עץ פורש מינימום:

1. נוכיח T עץ פורש: נוכיח כי T גרף קשיר ללא מעגלים, ושהוא תת גרף של הגרף המקורי (טריוויאלי כי $E_T \subseteq E$). כל שנותר הוא להוכיח קשיר +ללא מעגלים. נוכיח כי הוא קשיר וכן כי $|E_T| = |V| - 1$.

ראשית, נראה כי

$$|E_T| = |E_{T'} \cup \{e\}| = |E_{T'}| + 1$$

אם כן, T' הוא עפ"מ ולכן מתקיים בו $|E_{T'}| = |V/e| - 1 = |V| - 1 - 1 = |V| - 2$, כלומר $|E_{T'}| = |V| - 2$.

סה"כ, $|E_T| = |E_{T'}| + 1 = |V| - 2 + 1 = |V| - 1$, כנדרש.

כעת, נוכיח קשירות. יהיו $x, y \in V$. אם המסלול הפשוט (היחיד) E מ$F(y)$ $F(x) \to$ לא משתמש בuv כקודקוד פנימי, אזי אותו מסלול קיים גם בT'.

אחרת, במסלול uv כן קודקוד פנימי. החלפת הקודקוד uv במסלול הפשוט E בקשת $u \to_e v$, מייצרת מסלול פשוט מx לy בT.

מדוע? T' עפ"מ בG/e ובפרט קשיר. לכן שם, קיים המסלול $x \rightsquigarrow uv \rightsquigarrow y$, כאשר נסתכל ב$T$ חזרה, נוכל להסתכל על אותו מסלול בדיוק, בתוספת הקשת $u \to v$. כלומר $x \rightsquigarrow u \to v \rightsquigarrow y$, מסלול זה קיים גם ב$T$ כי לא שינינו דברים פרט לuv, ולכן סה"כ קיים מסלול פשוט בין x לy.

לכן T קשיר.

סה"כ T קשיר וכן $|E_T| = |V| - 1$ ולכן T עץ פורש.

2. נוכיח כי T הוא בעל משקל קטן ביותר: נניח בשלילה כי T לא עץ פורש מינימום. אזי קיים $\hat{T}$ עפ"מ עבור G. מלמהו, נניח בה"כ כי $\hat{T}$ מכיל את e (הקשת הקטנה ביותר, אשר קיים עפ"מ שמכיל אותה לפי הלמה. נניח שזה העץ). נכווץ את $\hat{T}$ על e ונקבל את $\hat{T}'$ שמוגדר:

$$\hat{T}' = (V/e, E_{\hat{T}'})$$

כאשר $E_{\hat{T}'} = E_{\hat{T}}/\{e\}$.

נשים לב כי $\hat{T}'$ הוא עפ"מ עבור G/e:

א. $|E_{\hat{T}'}| = |E_{\hat{T}}| - 1 = |V| - 1 - 1 = |V/e| - 1$

ב. וכן, יש להראות כי $\forall x, y \in V/e$ יש מסלול ב־$\hat{T}'$:

אם המסלול $\hat{P}$ מ־x ל־y ב־$\hat{T}$ משתמש ב־e אזי הוא נראה כך –

$$x \to u \to_e v \to y$$

לאחר הכיווץ על G/e מסלול זה ב־$\hat{T}'$ יראה כך: $x \to uv \to y$, כלומר מסלול זה הוא מסלול מ־x ל־y ב־$\hat{T}'$.

אחרת, המסלול $\hat{P}$ לא משתמש ב־e, לאחר הכיווץ הוא ישאר אותו מסלול בדיוק.

סה"כ, בכל מקרה קיים מסלול מ־x ל־y – לכן הגרף קשיר.

סה"כ $\hat{T}'$ קשיר + $|E_{\hat{T}'}| = |V/e| - 1$ ולכן $\hat{T}'$ הוא עץ פורש.

ג. נראה כי

$$w(\hat{T}') = w(\hat{T}) - w(e) < (*) w(T) - w(e) = (**) w(T')$$

$(*)$ כי מההנחה $\hat{T}$ הוא עפ"מ ולכן $w(\hat{T}) < w(T)$. $(**)$ כי לפי ההגדרה, T' הוא העץ שהורידו ממנו את e.

סה"כ, קיבלנו $\hat{T}'$ הוא עפ"מ על G/e, בפרט $w(\hat{T}') < w(T')$, בסתירה! כי T' הוא עפ"מ על G/e (מהנתון), ולכן $w(T') \leq w(\hat{T}')$.

מסקנה – T הוא בעל משקל קטן ביותר, וסה"כ T הוא MST. כנדרש.

$\blacksquare$.

0.8 האלגוריתם של פרים $(Prim)$

הרעיון באלגוריתם: להתמודד עם הבעיה עם איזה חתך נתחיל ונבחור בכל שלב?". ברעיון זה בוחרים חתך שבו בצד אחד קודקוד אחד, ובצד השני שאר הקודקודים. כמו באלגוריתם הגנרי, נרצה לבחור את הקשת הקלה ביותר (אם יש כמה באותו גודל – בוחרים אחת מהן).

באיטרציה הראשונה – בוחרים קודקוד שרירותי. לאחר מכן ממשיכים איתו עד הסוף, נניח שהקודקודים הינם $u_1, ..., u_n$. עם הקשתות $u_1 \to u_2$ וכו'. מכווצים את $u_1 \to u_2$. כעת מקבלים u_1u_2 מצד אחד, ומהצד השני של החתך שאר הקודקודים $u_3, ..., u_n$ וכך ממשיכים את האלגוריתם עד שמקבלים קודקוד יחיד $u_1u_2u_3..u_n$.

האלגוריתם הולך להשתמש בתור קדימויות, באשר יש לו שתי פעולות עיקריות: $Init, ExtractMin$.

האלגוריתם $(G=(V,E),w)$ **:**

1. $E_T=\emptyset$

2. עבור כל $u\in V$ בצע:

א. $u.key=\infty$ (המפתח יגיד את משקל הקשת **הקלה** ביותר בחתך שנוגעת בu ועוברת דרך החתך לצד השני - נשים לב: מחליפים את הערך של המפתח רק אם הערך קטן יותר מהערך הקודם שהופיע שם.)

ב. $u.\pi=null$ (הפאי יגיד לנו מי הקודקוד בצד השני של הקשת, שהמשקל שלה הוא $u.key$ - ושוב, נשים לב שערך הפאי ישתנה רק אם ערך הkey השתנה, וישתנה למי שמחזיק בערך זה.)

3. בחר קודקוד שרירותי $r\in V$ ואתחל $r.key=0$
4. $Q.init(V)$ - אתחל את תור הקדימויות.

5. כל עוד $|Q|\geq 1$ בצע:

א. $u=Q.extract.Min()$ (שקול לבחירת קשת קלה ביותר" - מוציאים אותו מהתור)

ב. אם $u.\pi\neq null$ אזי $Add(u,u.\pi)\rightarrow E_T$ (חשוב לשים לב - בפעם הראשונה שנכנס לסעיף 5 באלגוריתם, נכנס בהכרח במצב בו $u.\pi=null$, כיוון שבסעיף 2ב' אתחלנו את כולם להיות $null$, ולכן לא נוסיף כלום).

ג. עבור כל $v\in ADJ[u]$: (תעבור על השכנים של u)

אם $v\in Q$ (אם עדיין בתוך התור) וגם $w(u,v)<v.key$ אזי -

1. $v.key=w(u,v)$
2. $v.\pi=u$

6. החזר E_T.

נכונות האלגוריתם: נובעת מנכונות האלגוריתם הגנרי. בכל שלב מסמלצים" כיווץ על הקשת שבחרנו, ובכל שלב מסתכלים על החתך של כקודקוד שבחרנו כעת עם כל מה שמקודם, למול מה שנשאר. זה אלגוריתם שמאוד דומה לאלגוריתם הגנרי, ומשם נכונותו.

זמן הריצה:

ניתן לראות ששלבים $1-3$ עולים $O(|V|)$ זמן, שכן עוברים על כל הקודקודים.

שלב 4 - תלוי ב$Queue$.

שלב 5א' - גם כן, תלוי בסוג ה$Queue$. אבל, כמה פעמים נבצע הוצאה? $O(|V|)$ הוצאות, כי כל קודקוד יכול לצאת פעם אחת.

נשים לב, שכל קודקוד יכול לצאת מהתור לכל היותר פעם אחת. כאשר קודקוד יוצא מQ, האלגוריתם עובר על כל השכנים שלו, ובמקרה הגרוע כל שכן גורר עדכון מפתח אחד. לכן, שלב 5ג' יעלה $deg(u)$. אבל - כמה פעמים בכלל הם יכולים להקטין את המפתח שלהם (לשנות את ערך הkey, בהכרח להקטין לפי מה שהסברנו)? מס' הפעמים הוא $\sum_{u\in V}deg(u)=2|E|=O(|E|)$ שהרי בסכימת הדרגות כל קודקוד נספר פעמיים.

לסיכום - זמן הריצה תלוי בQ:

א. מערך: אתחול יעלה $O(|V|)$, ויתבצע פעם אחת. הוצאת המינימום תעלה $O(|V|)$, ותתבצע $|V|$ פעמים - וסה"כ תעלה $O(|V|^2)$, הקטנת מפתח תעלה $O(1)$ ותתבצע $2|E|$ פעמים. סה"כ סיבוכיות הזמן במערך תהיה $O(|V|^2+|V|+|E|)$, כמו כן $|E|\leq|V|^2$ ולכן סה"כ סיבוכיות זמן הריצה $O(|V|^2)$.

ב. ערימה בינארית/בינומית: אתחול יעלה $O(V)$ ויתבצע פעם אחת, הוצאת המינימום תעלה $O(log|V|)$ ותתבצע $|V|$ פעמים, וכן הקטנת מפתח תעלה $O(log|V|)$ ותתבצע $2|E|$ פעמים. סה"כ סיבוכיות הזמן בערימה תהיה $O(|V|log|V|+|V|+|E|log|V|)$ וכן, כיוון שהגרף קשיר מתקיים $|E|\geq|V|-1$ בפרט $|E|log|V|\geq|V|log|V|$ ולכן סה"כ סיבוכיות זמן הריצה הינה $O(|E|log|V|)$.

ג. ערימת פיבונאצ'י: אתחול ב$O(|V|)$, הוצאת מינימום $|V|$ פעמים שתעלה $log|V|$, וכן הפחתת מפתח עולה $O(1)$ לשיעורין שתתבצע $2|E|$ פעמים. לכן סה"כ $|V|+|V|log|V|+E$ וסיבוכיות זמן הריצה $O(|V|log|V|+|E|)$.

לא מומלץ להשתמש במערך. נשאלת השאלה מה עדיף - בערימה בינארית או בערימת פיבונאצ'י? תמיד מתקיים הרי כי $|V|\leq|E|$, ולכן ניתן לראות שעדיף להשתמש בערימת פיבונאצ'י בזמן ריצה של $O(|V|log|V|+|E|)$.

0.9 האלגוריתם של קרוסקל

הרעיון: נבחר בכל פעם את הקשת הקלה ביותר **בכל הגרף**, ונוסיף אותה לעץ פורש המינימום. (בפרט, היא תהיה הכי קלה בחתך כלשהו).

הקושי - לדאוג שאין מעגלים / אין לולאה עצמית בזמן הכיווץ.

להלן האלגוריתם:

MST-KRUSKAL(G=(V,E),w):
1. $E_T=\emptyset$
2. for each $u\in V$:
a. make_set(u)
3. for every edge $e=(u,v)\in E$ in increasing order of weights
4. if find_set(u) $\neq$ find_set(v)
a. Add (u,v) to E_T
b. union (u,v)
5. return E_T

כפי שניתן לראות - האלגוריתם משתמש ביוניון פיינד. בתחילה, לכל קודקוד ניצור קבוצה. לאחר מכן, נתחיל מהקשת הקלה ביותר, ונעלה בסדר עולה של משקלים, כך נעבור על כל הקשתות. בכל שלב, נבדוק האם שני הקודקודים שמרכיבים את הקשת נמצאים באותה קבוצה . אם לא - נוסיף את הקשת בניהם לקבוצת הקשתות, ונאחד בין הקבוצות שלהם (נשים לב שיתכן ונוצר מצב בו הקבוצות כרגע הם $\{u,v\},\{w,x\}$ ואנו נדרשים לבדוק את הקשת uw. אין בניהם כרגע קשת ולכן אנחנו נוסיף אותה לMST וכן נאחד בין הקבוצות לקבוצה גדולה $\{u,v,w,x\}$).

למעשה - מה שהאלגוריתם עושה קורה בשלבים $3-4$. אם שני האיברים זרים זה לזה ולא נמצאים באותה קבוצה, חייבים להוסיף קשת שתחבר בניהם בעץ, בשביל שיהיה עץ פורש ובפרט קשיר. אם הם כבר באותה קבוצה, אין צורך להוסיף קשת שתחבר בניהם. מהיכן מגיע המינימום? מהמעבר על הקשתות לפי הקשת הנמוכה ביותר עד לגדולה ביותר. בכל מקרה, נעבור על כל הקשתות - אך כשנגיע למצב שכל האיברים באותה קבוצה ויש קבוצה אחת - סיימנו.

כיצד האלגוריתם בוודאות לא יבחר מעגל? נניח ובחרנו קודקודים $u_1,...,u_k$. נרצה לעבור על קשת $u_1\to u_k$. אם נוסיף אותה, בהכרח יווצר מעגל. בשלב 4 אנחנו בדיוק

בודקים את זה ־ לאחר שלב 4 נקבל תשובה שהקודקודים באותה קבוצה ולכן לא נוסיף קשת זאת, וכך לא יווצר מעגל.

זמן הריצה:

לפי האלגוריתם ניתן לראות כי:

מבצעים $O(|V|)$ פעולות $makeSet$, כלומר $O(1)$ וסה"כ $O(|V|)$, וכן מבצעים $|V|-1$ פעולות $union$, שכל אחת עולה $O(log^*|V|)$ ולכן סה"כ $|V|log^*|V|$. כמו כן מבצעים $|E|$ פעמים $findSet$ שעולה $O(log^*|V|)$ ולכן סה"כ $O(|E|log^*|V|)$. כמו כן, עלינו למיין את הקשתות E, לכאורה ניתן למיין ב$|E|log|E|$ אך עם מעט יותר מידע על סוג המשקלים ניתן גם ב$O(|E|)$ זמן. סה"כ סיבוכיות זמן הריצה תהיה ־

$$O(|V| + (|E| + |V|)log^*|V| + sort(E)) = O(|E|log^*|V| + sort(E))$$

אם $Sort(E) = |E|$ אזי סיבוכיות זמן הריצה הינה $O(|E|log^*|V| + |E|)$, אחרת $O(|E|log^*|V| + |E|log|E|)$

בהשוואה בין זמני הריצה של פרים וקרוסקל ־ בדרך כלל קרוסקל ינצח. אך אם מס' הקשתות גדול יחסית, ממש גדול יחסית ־ אזי עדיף להשתמש בשל פרים. אחרת, של פרוסקל ינצח.