אלגוריתמים 1 - הרצאה 4

18 בנובמבר 2025

גיא יער־און

0.1 $SSSP$ בגרפים ממושקלים

קלט: גרף $G=(V,E)$, $s\in V$ ופונקציית משקל $w:E\to\mathbb{R}$.
פלט: $\forall v\in V:\delta(s,v)$

0.1.1 נסיון ראשון - תכנות דינמי

נזכר כי במקרה הממושקל:

$$\delta(u,v):=\left\{\begin{array}{cc} min_{p=u\rightsquigarrow v}\{w(p)\} & \exists p=u\rightsquigarrow v \\ \infty & o.w \end{array}\right\}$$

ראינו בלמה 1, שתת מסלול של מסלול קצר ביותר הוא גם כן מסלול קצר ביותר. זה נכון גם עבור גרפים ממושקלים. אם כן, נשים לב שנוכל לקבל כאן רעיון לאלגוריתם רקורסיבי: אם נרצה לחשב את המסלול הקצר ביותר מs אל u, וידוע כי שכניו של u הינם $u_1,..,u_4$ אזי נחשב את המסלול $s\to u_1$ למשל, ונוסיף משקל קשת. כלומר -

$$f(u)=min\{f(u_1)+w(u_1,u),f(u_2)+w(u_2,u),f(u_3)+w(u_3,u),f(u_4)+w(u_4,u)\}$$

נשים לב כי אם יש מסלול אל u, בפרט ישנו קצר ביותר, והוא בוודאות יעבור קודם לכן אצל אחד השכנים של u (לפי אותה למה). ובמילים אחרות: המסלול הקצר ביותר אל u חייב לעבור אצל אחד מהקודקודים השכנים של u.
נתבונן בנוסחה הבאה, באשר $f(u)=\delta(s,u)$:

$$f(u)=\left\{\begin{array}{cc} 0 & u=s \\ min\{min_{(v,u)\in E}\quad(\{f(v)+w(v,u)\}),\infty\} & o.w \end{array}\right\}$$

הנוסחה די ברורה, נשים לב לשני דברים:
א. אנחנו מבצעים השוואה בין הביטוי לבין אנסוף - יתכן שלא קיים מסלול בין s לu.
ב. נשים לב כי הגרף מכוון, אנחנו עוברים על הקשתות (v,u) כלומר הקשתות שנכנסות אל u.

הלוואי וזה היה עובד, זה היה קל מאוד. באלגוריתם הנ"ל יש בעיה. מה הבעיה?
נניח שיש מעגל wvu וישנה קשת $s\to v,s\to w$. כשנחשב ברקורסיה את המסלול הקצר ביותר מs אל v, נצטרך לחשב את $f(w)$ כי יש בניהם קשת, כשנחשב את $f(w)$ נגלה שאנחנו צריכים לחשב

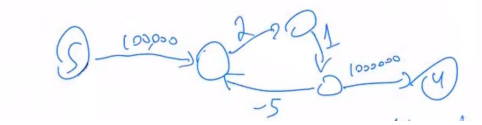
את $f(u)$ כי ישנה קשת $u \to w$, ואז שוב צריך לחשב את $f(u)$. וכשנצטרך לחשב אותו - חוזר חלילה. נתקע בלופ: הרקורסיה תקרוס, והבוס שלך יפטר אותך. באסה.

מסקנה: בגרף מכוון, ללא מעגלים, נוסחה זו תעבוד בסיבוכיות $O(|V|+|E|)$.

0.1.2 סוגי מעגלים

כפי שראינו, מעגלים עושים לנו בעיות. ישנם מספר סוגי מעגלים:

א. מעגל שלילי - מעגל כמו המעגל המתואר כאן לעיל הוא מעגל שלילי. הליכה על מעגל שכזה תמיד תהיה טובה לנו כי נעבוד $2+1$ וירד לנו -5 כלומר בכל סיבוב אנחנו נרוויח -2. במצב כזה, אם נלך אנסוף פעמים על המעגל נגיע למשקל שהוא מאוד נמוך - מינוס אנסוף.



לכן, אם יש מעגל שלילי מs לu בG נגדיר: $\delta(s,u) = -\infty$.

כעת, נניח כי בגרף יכולים להיות משקלים שליליים אך אין מעגלים שליליים.

ב. מעגל אפס - מעגל שסכום המשקלים שלו הוא אפס, ואין סיבה לכאורה לעבור בו. למשל מעגל כמו שמתואר מעלה, במקום 1 שיהיה הערך 3. נראה כי סכומו יהיה $2+3-5=0$. מעגל כזה לא מפריע לנו.

ג. מעגל חיובי - מעגל שסכום הערכים על הקשתות שלו חיובי. אין סיבה לעבור עליו במציאת מסלול קצר ביותר.

מסקנה: אם אין מעגלים שליליים, ניתן להניח שקיים מסלול קצר ביותר שהוא מסלול פשוט. (מדוע? כי לא נרצה לעבור במעגל חיובי, ועל מעגל אפס אפשר לדלג).

מכאן, ניתן להניח כי במסלול קצר ביותר יש לכל היותר $|V|-1$ קשתות. (זהו מסלול פשוט: אין בו מעגלים, ולכן על כל קודקוד ניתן לעבור לכל היותר פעם אחת. במקרה הגרוע ביותר עברנו על כל $|V|$ הקודקודים, יש בניהם $|V|-1$ קשתות).

נשים לב - גם אם אין מעגלים שליליים, אנחנו עדיין באותה בעיה שנתקלנו בה באשר ניסינו לעבוד בתכנות דינמי. אנחנו לא מפרקים את הבעיה לתתי בעיות שם - תמיד נתקע בלופ.

0.1.3 הקלת קשתות $-Relaxtions$

רעיון האלגוריתם: לכל $u \in V$ האלגוריתם מתחזק ערך $d[u]$ שהוא חסם עליון על $\delta(s,u)$. כלומר, $d[u] \geq \delta(s,u)$.

האלגו' על פני ריצתו יוריד ערכי d עד ש(בתקווה) כל ערכי הd מקיימים:

$$\forall u \in V : d[u] = \delta(s,u)$$

כמו כן, נשתמש במשתנה $\pi[u]$ להגדיר את עץ המסלולים הקצרים ביותר.

כך יראה האתחול:

```
INITIALIZE-SINGLE-SOURCE(G = (V, E), s)
for each vertex v ∈ V
    d[v] ← ∞
    π[v] ← NULL
d[s] ← 0
```

** נשים לב כי $d[s] = 0$ היא הנחה לגיטימית, כלומר $\delta(s,s) = 0$, הדרך היחידה ש$\delta(s,s) \neq 0$ היא שיהיה מעגל שלילי שמתחיל ונגמר בs, ואז $\delta(s,s) < 0$, אך **אנחנו מניחים שאין מעגלים שליליים.**

נשים לב: אנחנו עובדים לפי קונבציה שs מצביע אל $null$ בעץ, וכן יתכנו קודקודים נוספים שמצביעים אל $null$ (מי שלא נגישים אל s), איך נבדיל בניהם לבין s? הd שלהם יהיה אנסוף ו$d[s] = 0$.

למת אי שוויון המשולש (במקרה הממושקל): עבור $G = (V, E)$ עם פונקציית משקל $w : E \to \mathbb{R}$ וכן $s \in V$ ו$(u, v) \in E$ מתקיים:

$$\delta(s, v) \leq \delta(s, u) + w(u, v)$$

מסקנה: נניח כי אכן $d[u] \geq \delta(s, u)$, אזי

$$d[u] + w(u, v) \geq \delta(s, u) + w(u, v) \geq \delta(s, v)$$

אזי, אם $d[v] > d[u] + w(u, v)$ אזי ניתן להקטין את $d[v]$ להיות $d[u] + w(u, v)$. מדוע? כי במקרה זה, כיוון ש$d[u] + w(u, v) \geq \delta(s, v)$, עדיין אם נגדיר $d[v] = d[u] + w(u, v)$ יתקיים $d[v] \geq \delta(s, v)$ כפי שרצינו. פעולת הקטנה זו נקראת **פעולת הקלה.**

(עוד אינטואיציה: נניח כי $d[u] = 100, d[v] = 200$, וכן ישנה קשת $e : u \to v$ כך ש$w(e) = 30$, אכן מתקיים $d[v] > d[u] + w(u, v)$, ולכן כדאי לשפר את $d[v]$ ולהקטין אותו להיות המסלול של $d[u]$ ועוד הקשת ששווה 30, שכן ערך המסלול ירד.)

פסאודו לפעולת $relax$ - ההקלה:

```
RELAX((u, v), w)
if d[v] > d[u] + w(u, v)
    d[v] ← d[u] + w(u, v)
    π[v] ← u
```

נשים לב כי אם בחרנו לבצע הקלה, שינינו את אבא של v להיות u ולכן π משתנה.

0.1.4 אלגוריתם מבוסס הקלות והוכחת נכונות של בלמן פורד

אלגוריתם מבוסס הקלות הוא אלגוריתם שמאתחל ערכי d, π פעם אחת בעזרת קריאה ל$Initalize$ –

$Single-Source$, ולאחר מכן כל העדכונים לערכי d יתבצעו רק בעזרת פעולות של $relax$.

המטרה: להראות כי האלגוריתם מבצע סדרה של הקלות שמובילות לכך ש:

$$\forall u \in V : d[u] = \delta(s,u)$$

כלל הטענות הבאות יהיו נכונות לכל אלגוריתם מבוסס הקלות.

למה 7 (**למת חוסר המסלול**)**:** לאחר ביצוע $ISS(G,s)$ (אתחול) באלגוריתם מבוסס הקלות, אם אין מסלול מS ל$v \in V$ בG אזי תמיד מתקיים $\delta(s,v) = d[v] = \infty$.

הוכחה: ההוכחה נובעת ישירות מלמה 9 שכאן למטה. בעת האתחול יתבצע $d[v] = \infty$. כמו כן, אם אין מסלול מתקיים $\delta(s,v) = \infty$. מתקיים $d[v] = \delta(s,v)$ ולפי למה 9 מהרגע שזה מתקיים הערך לא ישתנה יותר, ואכן מתקיים כדרוש.

למה 8 (**למת הקלת מסלול**) **:** נניח כי המסלול $P = (v_0, ..., v_k)$ הוא מסלול קצר ביותר עם $s = v_0$. נניח שבזמן ריצת אלגוריתם מבוסס הקלות לאחר ביצוע $ISS(G,s)$, סדרת ההקלות שהאלגוריתם מבצע מכילה את סדרת הקשתות של p כתת סדרה לפי סדרן בP. (כלומר, לאחר ביצוע פעולות רילקס על כל (v_i, v_{i+1}) לכל $0 \leq i \leq k-1$ לפי הסדר **אך לא בהכרח ברצף**) אזי, בסיום ריצת האלגוריתם מתקיים $d[v_k] = \delta(s, v_k)$. (כלומר, הבעיה נפתרה עבור קודקוד v_k).

הוכחה: נראה כי לאחר ביצוע הקלה על הקשת הi (בתת הסדרה) מתקיים $d[v_i] = \delta(s, v_i)$. נוכיח באינדוקציה על i.

בסיס: $i = 0$, כלומר משמעות הדבר היא $v_0 = s$, אכן בתחילה מתקיים $d[s] = \delta(s,s) = 0$. ע"פ למה 9, הערך של $d[s]$ לא ישתנה עוד לעולם.

צעד: ע"פ הנחת האינדוקציה, אחרי ההקלה על הקשת הi בתת סדרה מתקיים שביצענו הקלות על הקשת הראשונה, השנייה וכו' עד הi ולכן מתקיים $\forall_{0 \leq k \leq i} : d[v_k] = \delta(s, v_k)$

בהקלה ה$i+1$, לפי למה 10, מתקיים כי $\delta[v_{i+1}] = \delta(s, v_{i+1})$, כי אכן ביצענו הקלה על כל הקשתות הקודמות ובפרט על זו הi, ולכן לאחר ביצוע הקלה יתקיים השוויון. כנדרש.

נשים לב. למה 8 טוענת שאם יש לי את המסלול הקצר ביותר, אזי אני יודע שפתרתי את הבעיה עבור v_k. מדוע צריך את זה? נסתכל על המסקנה הבאה ־

מסקנה: נשים לב כי לפי למה 8 ולפי למה 1, אם פתרנו את הבעיה עבור v_k, אזי פתרנו את הבעיה לכל $0 \leq i < k$, כלומר $\forall_{0 \leq i < k} : d[v_i] = \delta(s, v_i)$. כלומר, אם נריץ פעם אחת נפתור את הבעיה עבור v_1. אם נריץ פעמיים, נפתור את הבעיה עבור v_2. וכן אם נריץ עבור $|V|-1$ נקבל את הפתרון להבעיה.

למה 9 (**למת החסם העליון של** $d[v]$)**:** באלגוריתם מבוסס הקלות, לאחר ביצוע האתחול $ISS(G,s)$ לכל קודקוד $v \in V$ תמיד מתקיים $d[v] \geq \delta(s,v)$. בנוסף, מהרגע שבו האלגוריתם מציב $d[v] = \delta(s,v)$, הערך של $d[v]$ לא ישתנה יותר עד לסיום ריצת האלגוריתם.

הוכחה: נוכיח באינדוקציה על מס' פעולות ההקלה שהאלגוריתם מבצע. נסמנם n.

בסיס: $n = 0$, $\forall v \in V/s$ אם האלגוריתם לא ביצע עדיין הקלות מתקיים $d[v] = \infty \geq \delta(s,v)$. וכן עבור $v = s$ מתקיים $d[s] = 0 = d(s,s)$ לפי הגדרת האתחול, ומתקיים $\delta(s,s) = 0$ כי אין מעגלים שליליים.

צעד: נניח שלפני ביצוע הקלה $Relax((u,v),w)$ מתקיים שלכל $x \in V$ $d[x] \geq \delta(s,x)$.

אם פעולת ה$relax$ לא שינתה שום ערך של d, לפי הנחת האינדוקציה עדיין מתקיים כי : $\forall x \in V$ $d[x] \geq \delta(s,x)$.

אם פעולת ה$relax$ כן שינתה ערך של d, היא יכולה לשנות רק את הערך של $d[v]$, בפרט יתקיים כעת כי $d[v] = d[u] + w(u,v)$, מתקיים כי לפי הנחת האינדוקציה כי $d[u] \geq \delta(s,u)$, ולכן נקבל

$d[v] = d[u] + w(u,v) \geq \delta(s,u) + w(u,w) \geq (*)\delta(s,v)$, באשר $(*)$ מתקיים לפי אי שוויון המשולש. סה"כ $d[v] \geq \delta(s,v)$ כנדרש.

נשים לב כי עלינו להוכיח דבר נוסף, מהרגע שנציב ערך $d[v] = \delta(s,v)$ הערך של $d[v]$ לא ישתנה יותר. אם כן, כיוון שפעולות $relax$ רק מקטינות ערכי d, והוכחנו הרי כי $\forall u \in V : d[u] \geq \delta(s,u)$ אזי ברגע ש$d[u] = \delta(s,u)$ הערך של $d[u]$ לא יכול להשתנות יותר כי $d[u]$ לא יכול לגדול, ו$d[u]$ לא יכול להיות קטן יותר כי $d[u] \geq \delta(s,u)$, סה"כ ערכו לא משתנה.

למה 10 (**תכונת ההתכנסות**): נניח שהמסלול $s \rightsquigarrow u \to v$ הוא מסלול קצר ביותר מs לv. אזי, לאחר ביצוע $ISS(G,s)$ באלגוריתם מבוסס הקלות, מתקיים שאם $d[u] = \delta(s,u)$ מתישהו לפני ביצוע הקלה על הקשת (u,v), אזי לאחר ביצוע הקלה על הקשת מתקיים $d[v] = \delta(s,v)$. (כלומר, אם המסלול מ$s \rightsquigarrow u$ כבר ידוע לי, לאחר ביצוע הקלה אחת אנחנו נדע את $s \rightsquigarrow v$ כי הרי סה"כ מוסיפים הליכה על קשת אחת).

הוכחה: ע"פ למה 9, תמיד מתקיים $d[v] \geq \delta(s,v)$, נפצל למקרים.

א. אם לפני ביצוע ההקלה התקיים $d[v] = \delta(s,v)$ אזי סיימנו, כי הערך לא ישתנה יותר לפי למה 9.

ב. אחרת, כלומר $d[v] > \delta(s,v)$. מכאן, $d[v] > \delta(s,v) = \delta(s,u) + w(u,v) = d[u] + w(u,v)$ כלומר קיבלנו $d[v] > d[u] + w(u,v)$, זהו בדיוק התנאי שאנחנו בודקים בביצוע הקלה. התנאי הזה מתקיים, ולכן אנחנו מעדכנים לאחר ביצוע ההקלה את $d[v] = d[u] + w(u,v)$, כמו כן מתקיים $d[u] = \delta(s,u)$ ולכן

$$d[v] = d[u] + w(u,v) = \delta(s,u) + w(u,v) = \delta(s,v)$$

ומלמה 9, ערך $d[v]$ לא ישתנה עוד. כנדרש.

0.1.5 האלגוריתם של בלמן פורד

נשים לב כי לפי למה 8, אם נדע את המסלול הקצר ביותר מs אל v נוכל לדעת את $\delta(s,v)$. ובכן, לשם כך נצטרך לפתור את הבעיה עצמה - המסלול הקצר ביותר. עם זאת, נשים לב כי אם נבצע הקלה לכל קשתות הגרף - לכל קשת פעם אחת: בוודאות ביצענו הקלה על הקשת הראשונה במסלול הקצר ביותר, אך לא ידוע לנו באיזה שלב של ההקלות ביצענו עליה הקלה. כלומר שאנחנו יודעים כי $d[v_1] = \delta(s,v_1)$ לפי למה 8. אם נבצע זאת שוב, הקלות לכל הקשתות בגרף, כעת בוודאות ביצענו הקלה בקשת $v_1 \to v_2$ ולפי אותה למה נדע כי $d[v_2] = \delta(s,v_2)$. באופן דומה ומחזורי: נבצע את התהליך שוב ושוב, עד שנקבל לאחר k איטרציות - עבור כל קודקוד u שיש אליו מסלול קצר ביותר מs עם לכל היותר k קשתות, יתקיים $d[u] = \delta(s,u)$.

כמה איטרציות אנחנו צריכים? בהינתן ההנחה שאין מעגלים שליליים, לכל קודקוד $u \in V$ קיים מסלול קצר ביותר מs אל u שהוא מסלול פשוט, במסלול כזה יתקיים כי אורכו $|V| - 1 \geq$. ולכן נצטרך לבצע לכל היותר $|V| - 1$ איטרציות.

הערה. אנחנו תמיד נדע את $d[v_1]$ בהתחלה כי יתקיים התנאי של אי השוויון לפי האלגוריתם כי $d[s] = 0$, זה לא בהכרח יקרה עבור שאר הקודקודים כי נקבל אי שוויון של $\infty + w() < \infty$ שאיננו נכון בהכרח.

להלן האלגוריתם:

בתחילה, נאתחל. לאחר מכן במשך $|V| - 1$ פעמים אנחנו נעבור על כל הקשתות בגרף ונבצע הקלה.

```
BELLMAN-FORD(G = (V, E), w, s)
INITIALIZE-SINGLE-SOURCE(G = (V, E), s)
for i = 1 to |V| − 1
    for (u, v) ∈ E
        RELAX((u, v), w)
```

סיבוכיות זמן הריצה: $O(|V| \times |E|)$

נכונות האלגוריתם: נניח וקיים מסלול, אזי קיים גם מסלול קצר ביותר (אחרת ראינו כי הערך יהיה אנסוף כבר בהתחלה, ולא ישתנה), אנחנו יודעים לחשב את $d[v_1]$ שבמסלול לפי הערה כאן לעיל, ומשם לפי למה 10, לאחר ביצוע הקלה אחת אנחנו יודעים גם את $d[v_2] = \delta(s, v_2)$. וכן הלאה ממשיכים עם הקלה ואח"כ למה 10 ולבסוף אנו יודעים את $d[v]$.

נשים לב: בDAG ישנו מיון טופולוגי, ולכן ניתן להחליט שנעבור בסדר מסויים על הגרף. וכך, לא נצטרך בכל שלב באלגוריתם בשורה **3** לעבור על כל הקשתות, נוכל לעבור בכל קודקוד רק על הקודקודים שיוצאים ימינה. נשים לב כי לקודקוד הראשון יש אפשרויות ללכת רק ימינה וכן הלאה על ההבאים. נקבל כי סיבוכיות זמן הריצה היא $O(|E| + |V|)$.

0.2 האלגוריתם של דייקסטרה

0.2.1 הגדרת הבעיה ומבוא

קלט: גרף $G = (V, E)$ ממושקל עם פונקציית משקל $w : E \to \mathbb{R}^+$ (משקלים חיוביים בלבד), וקודקוד מקור $s \in V$

פלט: $\forall u \in V : \delta(s, u)$ וכן להחזיר עץ מסלולים קצר ביותר מs.

באלגוריתם של פרים, למציאת MST, בכל איטרציה האלגוריתם בוחר את הקשת בעלת המשקל הנמוך ביותר שחוצה את החתך ומוסיף אותה לעץ. האלגוריתם של דייקסטרה פועל באופן דומה. בכל איטרציה, האלגוריתם של דייקסטרה בוחר את הקשת הטובה ביותר להוסיף לעץ המסלולים. השוני בין האלגוריתמים - מה זה אומר "הטובה ביותר".

הגדרה: נסמן בS את קבוצת הקודקודים שנמצאת כרגע בעץ המסלולים הקצרים ביותר. מסלול P יקרא "מסלול מיוחד" אם כל הקודקודים במסלול P נמצאים בS חוץ מהקודקוד האחרון שלא נמצא בS. ייתכנו כמה מסלולים מיוחדים P כנ"ל.

האלגוריתם של דייקסטרה בכל איטרציה בוחר את הקודקוד u, $u \notin S$ כך שיש מסלול מיוחד מs לu שכרגע הוא מסלול מיוחד קצר ביותר ביחס לכל המסלולים המיוחדים. כלומר, אנחנו בוחרים קודקוד u שלא נמצא בתוך קבוצת הקודקודים בעץ, והקודקוד u הזה שאנחנו בוחרים הוא זה שהמסלול מs אליו הוא הקצר ביותר ביחס ל**כל המסלולים המיוחדים**. זוהי נקודה מהותית - הוא בוחר ביחס לכל המסלולים המיוחדים, לא ביחס לכל מי שמסתיים בu. האלגוריתם מוסיף את הקשת האחרונה על המסלול (זאת שנראית כרגע "הטובה ביותר") המיוחד P לעץ ומוסיף את u לS.

0.2.2 האלגוריתם

האלגוריתם של דייקסטרה הוא אלגוריתם מבוסס הקלות, לכן הוא מחזיק ערכי d וπ וכן מבצע ISS בתחילת הריצה ומוריד את ערכי d רק באמצעות $relax$.
להלן האלגוריתם:

```
DIJKSTRA(G = (V, E), w, s)
INITIALIZE-SINGLE-SOURCE(G = (V, E), s)
S ← ∅
Q.Init(V)
while Q ≠ ∅
    u ← Q.extract_min()
    Add u to S
    for each v ∈ ADJ[u]
        RELAX((u, v), w)
```

כאלגוריתם מבסס הקלות מתחילים ב$Iss(G,s)$ ויוצרים קבוצה ריקה S. כמו כן, אנו משתמשים בתור קדימויות Q. מאתחלים אותו ראשית. נגדיר כי מפתחות התור הם ערכי d. בכל שלב, כל עוד התור לא ריק (בתחילה הוא מאותחל עם כל איברי V כל הקודקודים), אנחנו נוציא את זה עם המפתח המינימלי (הd הכי קטן, המרחק הכי קטן..), נכניס אותו אל S, ונעבור על כל שכניו ונבצע להם הקלה.

ישנה נקודה שצריך לשים לב אליה - אם הערך של $d[v]$ יורד, משמעות הדבר היא כי צריך לבצע $decrease_key$ על v בQ.

הקודקוד הראשון שיצא מQ הוא s כי $d[s]=0$ ושל כל השאר הוא אנסוף. וממנו יבנה העץ.

סיבוכיות זמן הריצה: נראה כי יש לנו אתחול, ידוע כי הוא עולה $O(|V|)$ זמן. הלולאה מתבצעת $|V|$ פעמים, חלק 7 מתבצע $deg(u)$ פעמים ולכן כמכלול $\sum_{v\in V} deg(v) = O(|E|)$ מס' הפעמים שיכולה להתבצע שורה 8, וכן בכל אחד מפעולות ה$relax$ אנחנו עלולים לבצע $decrease_key$. מכאן ש:

ISS שיעלה $O(|V|)$
ביצוע $init$
$|V|$ פעמים $extract_min$
$|E|$ פעמים $decrease_key$
וזמן הריצה הוא:

$$O(init + |V| + |V| \times extract_min + |E| \times decrease_key)$$

ערימה בינארית:

$$O(|V| + |V| + |V| \times log(|V|) + |E| \times log(|V|)) = O(|V|log(|V|) + |E|log(|V|))$$

ערימת פיבונאצ'י:

$$O(|V| + |V| + |V| \times log(|V|) + |E| \times O(1))) = O(|V|log(|V|) + |E|)$$

0.2.3 הוכחת נכונות של דייקסטרה

האלגוריתם של דייקסטרה הוא אלגוריתם חמדני. וניתן להוכיח למה בחירה חמדנית וכו'. אך הפעם נשתמש בהוכחה יותר ישירה.

הערה. אינווריאנטה היא טענה שנרצה להוכיח שתמיד מתקיימת.

אינווריאנטה 1: כל פעם שהאלגוריתם של דייקסטרה מגיע לשורה 4 בפסודו קוד, אז לכל $u \in S$ מתקיים $d[u] = \delta(s,u)$ (כלומר, כל מי שהכנסנו כבר לקבוצה S ונכנס לעץ, אנחנו יודעים את המרחק הקצר ביותר עבורו. אם זה נכון, זה בפרט יהיה נכון כאשר נגיע לשורה 4 בפעם האחרונה, ואז יתקיים לכל הקודקודים כי $d[u] = \delta(u,s)$ והוכחנו את הטענה).

הוכחה: ההוכחה היא באינדוקציה על מס' הפעמים שהאלגוריתם בריצה מסויימת מגיע לשורה 4. נסמנו n.

בסיס: $n = 0$, בשלב זה כל מה שבוצע עד כה באלגוריתם היה אתחול בלבד. הקבוצה S ריקה ובפרט האינווריאנטה מתקיימת באופן ריק.

צעד: נראה שכל פעם שהאלגוריתם מוסיף קודקוד $u \to s$ מתקיים $d[u] = \delta(s,u)$. יחד עם למה 10, נקבל שמהרגע שבו u התווסף אל S תמיד יתקיים $d[u] = \delta(s,u)$.

נניח בשלילה כי u מתווסף אל S גדולה, אך $d[u] \neq \delta(s,u)$. בה"כ u הוא הקודקוד הראשון שמתווסף אל S שעבורו $d[u] \neq \delta(s,u)$. ולכן כל הקודקודים שקדמו לu, יקיימו $d[k] = \delta(s,k)$.

נשים לב כי $u \neq s$ כי כבר באתחול $d[s] = \delta(s,s) = 0$ (כי אין מעגלים שליליים כי אין קשתות עם משקל שלילי) ולפי למה 10 הערך לא ישתנה לעולם. מכאן שלא יתכן ש$u = s$.

מכאן נסיק כי לפני שu התווסף אל S, בוודאות $S \neq \emptyset$ (בוודאות s יהיה שם). כמו כן, בהכרח קיים מסלול מs אל u כי אחרת לפי למה 7 (למת חוסר המסלול) יתקיים כי $d[u] = \infty = \delta(s,u)$, ולכן בוודאות יש מסלול $s \rightsquigarrow u$. כלומר $\delta(s,u) < \infty$.

נסמן P' מסלול קצר ביותר מs אל u בG (קיים כזה), נסמן בy את הקודקוד הראשון בP' שנמצא בV/S. בהחלט יתכן כי $y = u$. כמו כן, נסמן את הקודקוד לפני y במסלול בx, ולכן $x \in S$. כלומר המסלול הינו $P' = (s, ..., x, y, ..., u)$

מכאן שבזמן שu הצטרף לS, אנו יודעים כי $d[x] = \delta(s,x)$ (כי הנחנו שu הוא הראשון עבורו אי השוויון קרה).

נסתכל על מה האלגוריתם עשה ברגע שx הצטרף אל S: האלגוריתם בשלב ההוא עבר על כל הקשתות שיוצאות מx ובפרט על הקשת (x,y) וביצע עליהן הקלות. מלמה 10 נובע, שלאחר הקלה על הקשת (x,y) נקבל כי $d[y] = \delta(s,y)$ (כי הרישא של P' שמסתיימת בy היא מסלול קצר ביותר מs אל u, ובפרט תת מסלול של מסלול קצר ביותר הוא מסלול קצר ביותר, ולכן המסלול עד x הוא הקצר ביותר, ומכאן לפי למה קודמת).

מכאן, $d[y] = \delta(s,y) \leq \delta(s,u) \leq d[u]$ (כי אנחנו ללא משקלים שליליים, והמסלול יכל רק לגדול)

מצד שני, האלגוריתם בחר בu להצטרף אל S כאשר y הוא אחת מהאופציות , לכן בשלב זה $d[u] \leq d[y]$ (אחרת y היה נבחר)

סה"כ קיבלנו $d[u] \leq d[y]$ וכן $d[y] \leq d[u]$ ומכאן $d[u] = d[y]$, כלומר $d[y] = \delta(s,y) \leq d[u] = d[y] \leq \delta(s,u)$ בפרט מתקיים כי כל השוונות באמצע מתקיימים ונקבל $d[u] = \delta(s,u)$, בסתירה לכך שהנחנו אי שוויון בניהם. כנדרש.

0.3 סיכום

ישנם שלושה אלגוריתמים שונים לבעיית $shorts\ paths$.

אלגוריתם BFS**:** מטפל ב$SSSP$ במקרה הלא ממושקל. סיבוכיות זמן הריצה שלו $O(|E|+|V|)$

האלגוריתם של בלמן פורד: מטפל ב$SSSP$ במקרה הממושקל. מניח שיתכנו משקלים שליליים אך לא יתכנו מעגלים שליליים. סיבוכיות זמן הריצה שלו $O(|E| \times |V|)$

האלגוריתם של דייקסטרה: מטפל ב$SSSP$ במקרה הממושקל, אך מניח שלא יתכנו משקלים שליליים. סיבוכיות זמן הריצה שלו $O(|V|log|V|+|E|)$ עם פיבונאצ'י ועם ערימה בינארית $O(|V|log(|V|)+|E|log(|V|))$.