אלגוריתמים - 234247

14 בפברואר 2010

תוכן עניינים

1 אלגוריתם BFS (Breadth First Search)

נתון - $G\left(V,E\right)$ לא מכוון וסופי, וצומת $s\in V$

המטרה - לחשב לכל צומת $v\in V$ את המרחק בין v ו-s.

סימון - $\delta\left(v\right)$ - המרחק בין s ל-v בגרף G. אם לא קיים מסלול בין s ו-v אז $\delta\left(v\right)=\infty$.

האלגוריתם ישתמש ב -

- $\lambda\left(v\right)$ - פלט האלגוריתם, לאורך המסלול הקצר ביותר בין s ל-v
- $pred\left(v\right)$ - ה"קודם" של v
- Q תור

1.1 האלגוריתם

1. לכל $v\in V\setminus\left\{ s\right\}$ בצע $\lambda\left(v\right)=\infty$, $pred\left(v\right)=NIL$
2. עבור s : $\lambda\left(s\right)=0$, $pred\left(s\right)=NIL$
3. הכנס את s לתור Q
4. כל עוד התור Q אינו ריק -
   (א) הוצא את הצומת בראש התור Q, נסמן u
   (ב) לכל שכן v של u עבורו $\lambda\left(v\right)=\infty$:
      i. $\lambda\left(v\right)=\lambda\left(u\right)+1$
      ii. $pred\left(v\right)=u$
      iii. הכנס את v לתוך התור Q

1.2 הוכחת נכונות

טענה 1.1 לכל צומת $v\in V$ עבורו $\lambda\left(v\right)\neq\infty$ בסיום ריצת האלגוריתם קיים בגרף G מסלול בין s ל-v באורך $\lambda\left(v\right)$.

הוכחה: באינדוקציה על ערך $\lambda\left(v\right)$ -

- בסיס - $\lambda\left(v\right)=0\Rightarrow v=s$ ואכן בגרף יש מסלול ריק בין s לעצמו, ואורכו אפס.
- צעד -
  - נניח נכונות לכל צומת v עבורו $\lambda\left(v\right)<k$ ונוכיח עבור כל הצמתים עבורם $\lambda\left(v\right)=k$.
  - אם $\lambda\left(v\right)=k$ אזי קיים צומת u עבורו $\lambda\left(u\right)=k-1$ וקשת $\left(u,v\right)\in E$.
  - מהנחת האינדוקציה נקבל כי קיים מסלול P בין s ו-u שארכו $k-1$. נשרשר לסוף המסלול P את הקשת $\left(u,v\right)$ וקיבלנו מסלול בין s ו-v שארכו k

■

מסקנה 1.2 לכל צומת $v \in V$ עבורו $\lambda(v) \neq \infty$ בסיום ריצת האלגוריתם מתקיים $\delta(v) \leq \lambda(v)$.

הערה 1.3 כל צומת נכנס לתור לכל היותר פעם אחת.

הערה 1.4 לכל צומת $\lambda(v) \neq \infty$ אמ"מ v נכנס מתישהו לתור Q

טענה 1.5 לכל שני צמתים u ו-v עבורם $\lambda(u) \neq \infty,\ \lambda(v) \neq \infty$ וגם $\lambda(u) < \lambda(v)$ אזי u נכנס לתור לפני v.

הוכחה: נראה שלכל צומת v שעבורו $\lambda(v) \neq \infty$ כל הצמתים u המקיימים $\lambda(u) < \lambda(v)$ בהכרח נכנסו לתור לפני הצומת v.

נוכיח באינדוקציה על ערך $\lambda(v)$ -

- **בסיס** - $\lambda(v) = 0 \Rightarrow v = s$ הטענה מתקיימת באופן ריק.
- **צעד** -
  - נניח נכונות עבור כל הצמתים v עם $\lambda(v) \leq k$ ונוכיח עבור צומת v המקיים - $\lambda(v) = k + 1$
  - כאשר v נכנס לתור Q, יוצא מ-Q צומת u, שהוא שכן של v המקיים $\lambda(u) = k$.
  - לפי הנחת האינדוקציה, כל הצמתים w עבורם $\lambda(w) < k$ נכנסו לתור לפני u. נתבונן על צומת $u \neq u'$ עבורו $\lambda(u') = k$. u' נכנס לתור לפני v כיוון שהצומת שעדכן את $\lambda(u')$ מאינסוף להיות k הוא בעל $\lambda = k - 1$ ולכן נכנס לתור לפני u.

■

משפט 1.6 לכל צומת v עבורו $\lambda(v) \neq \infty$ בסיום ריצת האלגוריתם, מתקיים - $\lambda(v) = \delta(v)$.

הוכחה: באינדוקציה על $\delta(v)$ -

- **בסיס** - $\delta(v) = 0$ גורר בהכרח $v = s$ ואכן האלגוריתם מסמן רק את $\lambda(s)$ ב-0.
- **צעד** - נניח נכונות עבור כל הצמתים v שמקיימים $\delta(v) < k$ ונוכיח עבור צומת v המקיים $\delta(v) = k$.
  - נתבונן על המסלול הקצר ביותר בגרף G בין s ו-v -

$$s \to v_1 \to v_2 \to \cdots \to v_{k-1} \to v_k = v$$

לפי טענה 1.5, הצומת v_{k-1} נכנס ל-Q לפני v, נסתכל על השלב בריצת האלגוריתם בו v_{k-1} יוצא מהתור -

  * אם $\lambda(v) = \infty$ אזי האלגוריתם יסמן $\lambda(v) = k$
  * אחרת - $\lambda(v) \neq \infty$ (כלומר v נמצא ב-Q או שכבר יצא ממנו) ולכן הצומת u שסימן את $\lambda(v)$ היה לכל היותר בעל ערך $\lambda(u) = k - 1$, לכן $\lambda(v) \leq k$.

כלומר בכל מקרה $\lambda(v) \leq k$.

– **מטענה 1.1** $\lambda(v) \geq \delta(v) = k$ **לכן בסך הכל** $\lambda(v) = \delta(v) = k$

■

טענה 1.7 לכל צומת v אם $\lambda(v) = \infty$ בסיום ריצת האלגוריתם אזי לא קיים בגרף G מסלול בין s ל-v.

הוכחה: נניח בשלילה שיש מסלול בין s ו-v בגרף G -

$$s \to v_1 \to \cdots \to \mathbf{w} \to \mathbf{u} \to \cdots \to v$$

נתבונן על הצומת האחרון במסלול שעבורו $\lambda(v) \neq \infty$ (עבור s זה מתקיים, לכן חייב להיות כזה) ונסמנו w, ונסמן את הצומת הבא אחריו ב-u. זה לא ייתכן כי בשלב בו w יצא מהתור Q עברנו על כל השכנים של w, ובפרט על u ולכן $\lambda(u) = \lambda(w) + 1$ ■

1.3 סיבוכיות

- אתחול - $O(|V|)$
- כל צומת נכנס לתור לכל היותר פעם אחת. לכן הכנסה לתור $O(|V|)$
- כשמוציאים צמתים מהתור, עוברים בסך הכל על כל קשת לכל היותר פעמיים (על הקשת (u, v) פעם אחת כאשר u יצא מהתור ופעם אחת כאשר v יצא מהתור) לכן הוצאה מהתור $O(|V| + |E|)$

בסך הכל - $O(|V| + |E|)$.

1.4 ה"קודמים" predecetor

נגדיר $G_{pred} = (V_{pred}, E_{pred})$, כאשר -

$V_{pred} = \{v \in V \mid pred(v) \neq NIL\} \cup \{s\}$

$E_{pred} = \{(pred(v), v) \in E \mid pred(v) \neq NIL\}$

- G_{pred} הוא תת גרף של G
- V_{pred} הוא קבוצת הצמתים ב-G שיש מסלול בינן ובין s
- G_{pred} קשיר
- G_{pred} הוא עץ (כי לכל צומת בו פרט ל-s יש בדיוק קשת אחת, לכן $|E| = |V| - 1$ וכאמור הוא קשיר)

1.5 הערות[1]

כאמור BFS הוא חיפוש לרוחב, לאיזה שימושים נוספים נוכל להשתמש בו?

[1] הרצאה שניה - 26.10.09

1.5.1 גרף דו צדדי

נתון גרף $G(V,E)$, האם G דו צדדי?

כלומר האם ניתן למצא V_1,V_2 כך ש-

$$\begin{aligned} V_1 \cap V_2 &= \varnothing \\ V_1 \cup V_2 &= V \end{aligned}$$

וכל קשת בגרף מקשרת בין צומת ב-V_1 וצומת ב-V_2.

טענה 1.8 נניח כי G קשיר, נריץ BFS מצומת מקור שייבחר שרירותית. אם יש בגרף שני צמתים שכנים x,y (כלומר $(x,y)\in E$) כך ש-$\lambda(x)=\lambda(y)$ אזי הגרף <u>אינו דו צדדי</u>.

לפני שנוכיח את הטענה נראה מדוע אם התנאי לא מתקיים אזי הגרף דו צדדי. נחלק את הצמתים לשתי קבוצות שייקבעו לפי הזוגיות של המרחק שלהן מצומת המקור, כלומר -

$$\begin{aligned} V_1 &= \{v|\lambda(v) \ \ mod2=0\} \\ V_2 &= \{v|\lambda(v) \ \ mod2=1\} \end{aligned}$$

ברור כי $V_1\cap V_2$ הוא קבוצה ריקה, וכיוון ש-G קשיר ברור כי $V_1\cup V_2=V$.

מדוע לא תיתכן קשת בתוך V_1 או V_2?

נניח בשלילה כי קיימת קשת (x,y) בתוך אחת הקבוצות -

- לא ייתכן כי $\lambda(x)=\lambda(y)$ - זו ההנחה שלנו.
- לכן - $n\in\mathbb{N}$ $\lambda(x)=\lambda(y)+2n$, אבל גם זה לא ייתכן כי אז יש מסלול מ-s ל-y באורך $\lambda(x)+1<\lambda(y)$.

כלומר לא קיימת קשת בתוך אותה קבוצה.

הוכחה: (לטענה 1.8) קיימים שני צמתים x,y כך ש-$\lambda(x)=\lambda(y)=d$ וקיימת קשת $(x,y)\in E$. הקשת (x,y) איננה נמצאת בעץ G_{pred} כי לא ייתכן שגילינו את x באמצעות y או את y באמצעות x.

אם כך יש בגרף G מעגל באורך אי זוגי - כיוון שמנקודת הפיצול (המסלול בין s ו-x, והמסלול בין s ו-y) יש מסלול באורך זהה d' עד ל-x ועד ל-y, ואל שני המסלולים האלו נוסיף את הקשת (x,y).

אם בגרף יש מעגל אי זוגי - לא ייתכן שהגרף אי זוגי. אפשר להראות בקלות - למשל על ידי צביעת כל הצמתים ב-V_1 בצבע אחד והצמתים ב-V_2 בצבע שני, נצבע צומת אחד במעגל בצבע מסויים - צבע זה קובע את הצבעים של כל אחד מהצמתים האחרים במעגל - ולבסוף צריך לצבוע את הצומת הראשון בצבע השני - סתירה! ■

2 אלגוריתם DFS (Depth First Search)

ב-BFS "פרסנו" את הגרף לפרוסות לפי המרחק מהשורש - ובכל פעם טיפלנו בפרוסה אחת. באלגוריתם DFS נרצה להעמיק ככל שניתן, ורק כשניתקע "נחזור אחורה" ונגלה צמתים אחרים.

"מרכז הפעילות" יתקדם בכל פעם שנגלה צומת חדש - ויחזור אחורה ל"צומת האב" רק כשלא נותרו עוד צמתים חדשים לגלות מתוך "מרכז הפעילות". התהליך יסתיים כשמרכז הפעילות יסוג לצומת ההתחלתי ולא יהיו צמתים נוספים שלא התגלו.

2.1 האלגוריתם

2.1.1 משתנים -

$k\left(v\right)$ מספר הצומת (שמו החדש)

$f\left(v\right)$ אב הצומת

i אינדקס רץ ($i \leq |V|$, טבעי)

2.1.2 האלגוריתם עצמו -

- אתחול -
  - $s \to v$; $1 \to k\left(s\right)$; $2 \to i$;
  - לכל $e \in E$ סמן e "חדשה"
  - לכל $u \in V \setminus \{s\}$ בצע $0 \to k\left(u\right)$ (סימון 0 פירושו צומת "חדש")
  - לכל $u \in V$ סמן $f\left(u\right)$ "בלתי מוגדר" (NIL)
- כל עוד ל-v יש קשת חדשה או $f\left(v\right)$ מוגדר, בצע -
  - אם יש ל-v קשת חדשה $e : v \to u$ אזי בצע -
    * סמן e ישנה
    * אם $k\left(u\right) = 0$ (כלומר u צומת חדש) בצע -
      · $i \to k\left(u\right)$; $v \to f\left(u\right)$;
      · $i + 1 \to i$
      · $u \to v$
  - אחרת ($f\left(v\right)$ מוגדרת) בצע - $f\left(v\right) \to v$

הערה 2.1 בגרף לא מכוון נפעיל את האלגוריתם שוב עם צומת s_2 שעדיין לא ביקרנו בו - כל עוד קיים צומת כזה.

2.2 סיבוכיות

- אתחול - $O\left(|E| + |V|\right)$
- ריצה -

טענה 2.2 זמן הריצה הוא לינארי, כלומר $O\left(|E| + |V|\right)$.

הוכחה: נניח שהגרף מיוצג על ידי רשימת שכנויות.

עבור צומת v קיימת רשימת הקשתות היוצאות ממנו, כאשר מרכז הפעילות מגיע לצומת ניתן לו את הסימון המתאים (סיבוכיות - $O\left(1\right)$), עבור כל קשת שמובילה לצומת שאנחנו כבר מכירים - נסמן את הקשת כמוכרת (בסיבוכיות - $O\left(1\right)$).

עבור קשת שמובילה לצומת חדש - נכנס אליו, ובסיום העבודה עליו נמשיך לעבור על רשימת הקשתות של v מאותו המקום שבו עצרנו - בצורה כזו מספר הצעדים שנבצע עבור כל צומת יהיה $O\left(N_v\right)$ כאשר N_v מספר השכנים של v. בסך הכל עוברים על כל צומת פעם אחת, ובכל צומת מבצעים מספר פעולות על פי מספר הקשתות היוצאות ממנו, ובסך הכל לכל הצמתים $\sum_{v\in V} N_v = |E|$ לכן סיבוכיות הריצה - $O\left(|V|+|E|\right)$. ■

2.3 סוגי הקשתות בגרף

- קשתות עץ - הקשתות דרכן DFS מגלה את הצמתים בפעם הראשונה
- קשתות קדמיות - קשתות (u,v) כך ש-u **אב קדמון** של v, לכן $k(u) < k(v)+1$, ו-v התגלה על ידי קשת אחרת (u',v).
- קשתות אחוריות - קשת (u,v) כך ש-v **אב קדמון** של u, לכן - $k(v) < k(u)$.
- קשתות חוצות - קשת (u,v) כך ש-u ו-v נמצאים בשני עצים שונים ביער ה-DFS שנוצר על ידי V וקשתות העץ.

2.4 משפטים

משפט 2.3 אם יש בגרף מסלול מכוון מ-s לצומת v אזי DFS יגיע ל-v.

הוכחה: נניח שהמסלול מ-s ל-v הוא -

$$s \to v_1 \to v_2 \to \dots v_{k-1} \to v_k = v$$

נניח בשלילה כי DFS לא הגיע ל-$v = v_k$, לכן בהכרח הוא לא הגיע ל-v_{k-1} (אם הוא היה מגיע אליו, הוא היה מסתכל על הקשת (v_{k-1},v_k) ומגלה את הצומת v_k) באותו האופן "מרכז הפעילות" לא הגיע ל-v_{k-2}, v_{k-3} וכו'.. וגם לא הגיע ל-s, אבל מרכז הפעילות היה ב-s - סתירה! ■

מסקנה 2.4 בהנתן גרף מכוון G ומקור s אלגוריתם DFS יגיע לכל הצמתים ה"ישיגים" מ-s.

טענה 2.5 יהא v צומת, ונניח שהצמתים שהתגלו לאחר שמרכז הפעילות הגיע ל-v ועד שמרכז הפעילות נסוג מ-v הם $\{u_1,u_2,\dots u_k\}$ על פי הסדר הזה.
אזי $u_1,u_2,\dots u_k$ צאצאים של v בעץ DFS.

הוכחה: באינדוקציה על מספר הצמתים k -

- בסיס - $k=1$, u_1 חייב להיות שכן של v, כי מרכז הפעילות עבר אליו מ-v וחזר אליו לאחר מכן.
- צעד - נניח כי $u_1,u_2,\dots u_{k-1}$ הם צאצאים של v, ונוכיח עבור u_k -

1. u_k מתגלה - u_k צאצא של u_{k-1}, וכיוון ש-u_{k-1} צאצא של v גם u_k צאצא של v.
2. אם u_k לא מתגלה ומרכז הפעילות נסוג ל-$f\left(u_{k-1}\right)$ התהליך נמשך לגבי $f\left(u_{k-1}\right)$ -

(א) או ש-u_k מתגלה על ידי $f\left(u_{k-1}\right)$, ואז u_k צאצא של $f\left(u_{k-1}\right)$, ולפי הנחת האינדוקציה $f\left(u_{k-1}\right)$ הוא v או צאצא של v, ולכן u_k צאצא של v.

(ב) אחרת מרכז הפעילות חוזר ל-$f\left(f\left(u_{k-1}\right)\right)$ וחוזר חלילה עד ש-u_k מתגלה או שמרכז הפעילות מגיע ל-v. אם u_k מתגלה - הוא צאצא של v, אחרת - u_k לא יתגלה עד לנסיגה מ-v, בסתירה לנתון.

■

משפט 2.6 [2] בעץ DFS מכוון, צומת v הוא צאצא של צומת u אמ"מ כאשר u סומן לראשונה ע"י DFS ניתן היה להגיע ל-v מ-u במסלול שמכיל רק צמתים לא מסומנים.

הוכחה: נחלק כרגיל לשני כיוונים -

- כיוון אחד - נניח ש-v צאצא של u בעץ ה-DFS.

$$u \to v_1 \to \cdots \to v_n \to v$$

ברור שכל הצמתים במסלול מ-u ל-v במסלול זה עדיין לא התגלו כאשר u סומן, כל הקשתות במסלול הזה הינן קשתות עץ - כלומר קשתות דרכן גילינו את הצמתים בפעם הראשונה.

- כיוון שני - נניח בשלילה ש-v איננו צאצא של u בעץ ה-DFS, אבל כאשר u התגלה היה בגרף מסלול מכוון מ-u ל-v שמכיל רק צמתים לא מסומנים.

$$u \to v_1 \to \cdots \to v_n \to v$$

בה"כ ניתן להניח ש-v הצומת הראשון במסלול הזה שיש לו התכונה הזו (כלומר איננו צאצא של u) ולהוכיח אינדוקטיבית לכל הצמתים במסלול, ונסמן w - הצומת הקודם ל-v במסלול הנתון מ-u ל-v.

הוכחנו כבר (טענה 2.5) כי אם הפעם הראשונה שמרכז הפעילות ל-v אחרי שהגיע ל-u ולפני הנסיגה מ-u, אזי v צאצא של u.

הנחנו בשלילה כי v לא צאצא של u, כמו כן ידוע לנו ש-v התגלה אחרי u.

מסקנה v התגלה אחרי הנסיגה של מרכז הפעילות מ-u, אבל כיוון שיש קשת מכוונת $v \to w$ מרכז הפעילות לא היה יכול לסגת מ-w (נסיגה שמתבצעת לפני הנסיגה מ-u) בלי לבחון את הקשת $v \to w$ - כלומר לגלות את v וזו סתירה.

■

2.4.1 קשתות עץ

מדוע "קשתות העץ" אכן משרות עץ מכוון מהשורש s?

- הקשתות האלו משרות מסלולים מ-s לכל הצמתים שגילינו במהלך ריצת DFS.
- דרגת הכניסה של כל צומת היא 1
- דרגת הכניסה של השורש היא אפס

ממשפטים שראינו בקומבי ברור כי הגרף המדובר הוא עץ.

[2] הרצאה 3 2.11.09

2.5 אלגוריתם DFS בגרף לא מכוון

טענה 2.7 אם קשת (a,b) בגרף לא שייכת לעץ DFS אזי a צאצא של b, או b צאצא של a.

הוכחה: נניח כי a התגלה לפני b כלומר $k(a) < k(b)$. תנאי משפט 2.6 מתקיימים, כאשר מרכז הפעילות הגיע ל-a בפעם הראשונה היה מסלול מ-a ל-b כשכל הצמתים בו לא מסומנים (המסלול הוא הקשת (a,b)) לכן לפי המשפט b צאצא של a. ■

מסקנה 2.8 בגרפים לא מכוונים יש רק שני סוגי קשתות -

- קשתות עץ
- קשתות אחוריות

מטענה 2.7 לא קיימות -

- קשתות חוצות - כי אם קיימת קשת (a,b) אזי a צאצא של b או להיפך.
- קשתות קדמיות - אם (a,b) איננה קשת עץ, ו-a צאצא של b נסמן את הקשת (a,b) כשנגיע ל-a, ולכן (a,b) קשת אחורית.

<u>שאלה</u> - נניח כי נתון לנו גרף קשיר. האם DFS יבקר בכל הצמתים בגרף כאשר הוא מתחיל משורש נתון s?

<u>תשובה</u> - כן. כיוון שהגרף קשיר יש מסלול מ-s (בהתחלה) לכל צומת בגרף שעובר רק בצמתים לא מסומנים ולכן לפי משפט 2.6 אלגוריתם DFS יבקר בכולם, וכולם יהיו צאצאים של s בגרף.

2.6 צמתי הפרדה ורכיבים אי פריקים

הגדרה 2.9 צומת s בגרף לא מכוון נקרא צומת הפרדה אם יש זוג צמתים a,b כך שכל המסלולים מ-a ל-b עוברים דרך s.

2.6.1 גרף פריק

הגדרה 2.10 גרף פריק הוא גרף קשיר שמכיל צומת הפרדה

2.6.2 רכיב אי פריק

הגדרה 2.11 תת גרף מושרה[3] שמקיים -

1. אי פריק (כגרף בפני עצמו)
2. אין תת גרף מושרה שמכיל אותו וגם הוא אי פריק.

[3] בהנתן גרף $G=(V,E)$, תת גרף $G'(V',E')$ מושרה של G אם -
1. $V' \subseteq V$
2. $E' = \{(u,v)\,|u,v \in V' \quad (u,v) \in E\}$

צומת הפרדה נמצא בחיתוך של שני (או יותר) רכיבים אי פריקים. בהנתן גרף פריק G נגדיר גרף על $\tilde{G}$. נסמן -

- $C_1, C_2, \ldots C_k$ - רכיבים אי פריקים של G
- $s_1, s_2 \ldots s_l$ - צמתי ההפרדה ב-G

ואז נגדיר -

- $\tilde{V} = \{C_1, C_2 \ldots C_k, s_1, s_2, \ldots, s_l\}$
- $\tilde{E} = \{(s_i, C_j) \,|\, s_i \in C_j\}$

טענה 2.12 עבור גרף פריק G, גרף העל $\tilde{G}$ הוא עץ

הוכחה: ברור שאם G קשיר גם $\tilde{G}$ קשיר. נניח בשלילה כי קיים מעגל ב-$\tilde{G}$ -
אם יש מעגל המכיל צמתי הפרדה אזי צמתי ההפרדה במעגל אינם יכולים להיות צמתי הפרדה בגרף המקורי. אבל $\tilde{G}$ גרף דו צדדי, ולכן בכל מסלול (בפרט מעגל) יש צמתי הפרדה, כלומר קיבלנו סתירה ו-$\tilde{G}$ קשיר חסר מעגלים, לכן עץ. ■

2.6.3 ה-lowpoint

הגדרה 2.13 לכל צומת $v \in V$ נגדיר $L(v)$ ה-$lowpoint$ של v באופן הבא, נתבונן בקבוצת הצמתים S המתקבלת ביחס לריצת DFS ספציפית -

1. צומת v
2. כל הצמתים שניתן להגיע אליהם על ידי הליכה קדימה בעץ DFS החל מ-v + קשת אחורית יחידה.

ונגדיר - $L(v) = \min_{u \in S} \{k(u)\}$.

טענה 2.14 נניח כי $k(u) > 1$, וקיימת קשת $v \to u$ בעץ DFS ו-$L(v) \geq k(u)$ אזי u צומת הפרדה.

הוכחה: $L(v) \geq k(u)$ לא יכולה להיות קשת אחורית מתחת העץ של v אל אחד האבות הקדמונים של u אחרת היה מתקיים $L(v) < k(u)$ בניגוד לנתון. ולכן u מפריד בין v לבין $f(u)$ (שמוגדר כי $k(u) > 1$) ■

טענה 2.15 אם u צומת הפרדה ו-$k(u) > 1$ אזי יש קשת $v \to u$ כך שמתקיים $L(v) \geq k(u)$

הוכחה: u צומת הפרדה, כלומר הגרף פריק, נסמן $H_1, H_2 \ldots H_n$ את הרכיבים האי פריקים הכוללים את u, כך ש-DFS התחיל ב-H_1, הגיע ל-u ואחר כך עבר ל-H_2 והגיע ל-$v \in H_2$ (בה"כ) כלומר מתקיים $L(v) \geq k(u)$ אחרת u לא היה צומת הפרדה. ■

כלומר קיבלנו איפיון (אמ"מ) לצומת הפרדה שאיננו שורש.

טענה 2.16 השורש s הוא צומת הפרדה אמ"מ יש לו לפחות שני בנים בעץ DFS.

הוכחה: שני כיוונים -

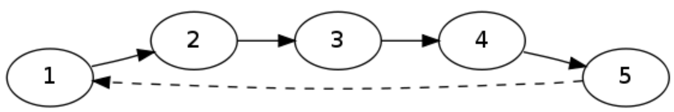
- כיוון אחד - נניח כי ל-s יש שני בנים u,v. s מפריד בין u ו-v כי אין בגרף DFS לא מכוון קשתות חוצות.
- כיוון שני - נניח כי s צומת הפרדה בין u ל-v. בה"כ DFS מבקר קודם כל ב-v, לכן DFS ייסוג מ-v רק אחרי שהוא ביקר בכל הצמתים ברכיב הקשירות H_1 ($v \in H_1$), רק אז (לאחר הנסיגה אל s) DFS יבקר ב-$u \in H_2$. כלומר ל-s יש לפחות שני בנים.

■

[4]<u>תזכורת </u>- אם u הוא לא שורש אזי u צומת הפרדה אמ"מ קיים ל-u בן v כך ש-$L(v) \geq k(u)$. אם u הוא שורש - u הוא צומת הפרדה אמ"מ יש לו לפחות שני בנים.

איך מחשבים את $L(v)$?

- כאשר צומת v מתגלה, $L(v) \leftarrow k(v)$
- כאשר DFS מגלה קשת אחורית $u \to v$ נעדכן - $L(v) \leftarrow min\{L(v), k(u)\}$.
- אם v לא השורש ו-u הבן של v בעץ, כאשר DFS נסוג מ-u ל-v בעץ נעדכן - $L(v) \leftarrow min\{L(v), L(u)\}$.



איור 1: דוגמא לחישוב lowpoint. הערך הסופי של $L(v)$ נקבע לאחר הנסיגה מ-v ל-$f(v)$, במקרה זה $L(v)=1$ לכל צומת בגרף.

2.6.4 ניתוח סיבוכיות

כאשר אנחנו רוצים לחשב lowpoint לכל צומת במהלך ריצת DFS נוסיף פעולות ב-$O(1)$ לכל פעולה (גילוי צומת, גילוי קשת אחורית, נסיגה מצומת). לכן סיבוכיות DFS נשמרת - $O(|V|+|E|)$.

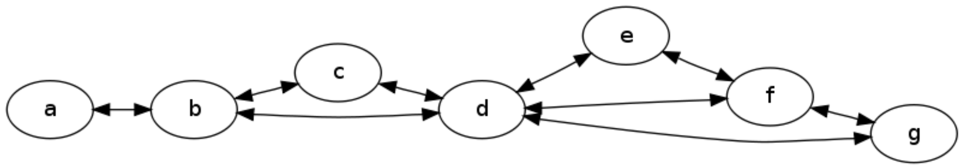
2.6.5 ניתוח ריצת DFS עבור גרף פריק

אלגוריתם DFS מגלה את הרכיבים האי-פריקים בגרף, ראשית הוא מגלה את הרכיבים האי פריקים שהם עלים בגרף העל של G, לאחר מכן הוא "מסיר" אותם מגרף העל וממשיך לגלות ולהסיר עוד ועוד רכיבים אי פריקים מגרף העל עד לסיום. לדוגמא -

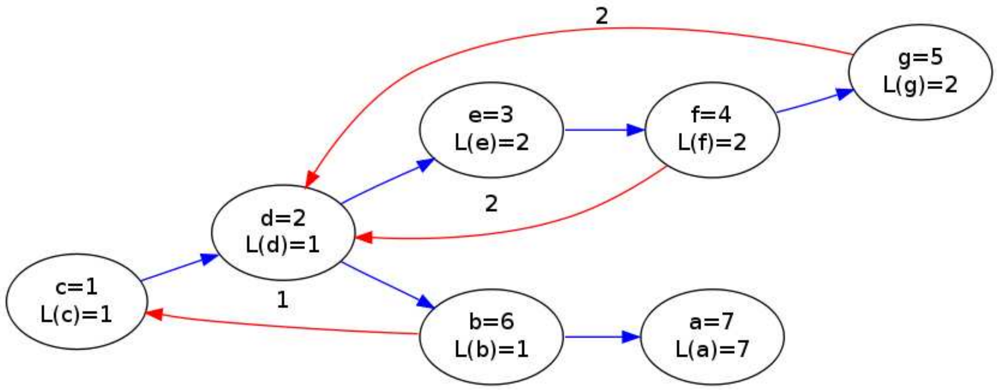
3 עץ פורש מינימום

הגדרה 3.1 נניח שנתונות תחנות בסיס - $b_1, b_2, \ldots b_n$ ויש זוגות של תחנות שיכולות לדבר בינהן - (b_i, b_j). נניח כי יש צומת r שרוצה לשדר לכל תחנות הבסיס, דרך מקובלת לעשות זאת הוא למצא עץ שהוא תת גרף של G ומכיל את כל צמתי הגרף והקשתות שלו יהיו קשתות בגרף. עץ כזה נקרא <u>עץ פורש</u> של הגרף.

[4]הרצאה 4 9.11.09



איור 2: הגרף שעליו מריצים DFS החל מהצומת c



איור 3: ולאחר הריצה (קשתות עץ בכחול, קשתות אחוריות באדום)

נניח כי לכל קשת בגרף יש מחיר אי שלילי, כלומר קיימת פונקציית משקל -

$$W : E \to \mathbb{R}^{+,0}$$

בהנתן עץ פורש T נגדיר -

$$w\left(T\right) = \sum_{e \in T} W\left(e\right)$$

המטרה - למצא עץ פורש T של גרף לא מכוון G עם פונקציית משקל W כך ש-$w\left(T\right)$ יהיה במינימום.

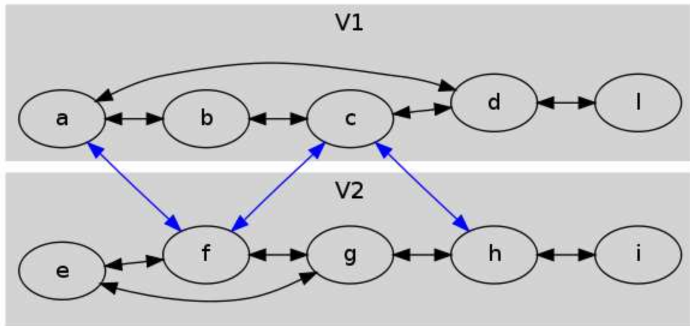
הגדרה 3.2 אם ניקח גרף G ונחלק אותו לשני חלקים - V_1, V_2, קשתות שעוברות בין V_1 ל-V_2 ייקראו <u>קשתות חתך</u>.

נשים לב שכל עץ פורש חייב להכיל לפחות קשת אחת מכל חתך.

- <u>הכלל הכחול</u> אם יש חתך שלא מכיל אף קשת כחולה (של ה"עץ" שנבנה איטרטיבית), צבע בכחול את קשת החתך עם המשקל המינימלי.
- <u>הכלל האדום</u> בהנתן מעגל, בחר את הקשת הכבדה ביותר (הלא צבועה) וצבע אותה באדום.

3.1 האלגוריתם הגנרי

- תחילה, כל קשתות הגרף לא צבועות.



איור 4: דוגמא לחתך, קשתות חתך בכחול

- צבע את הקשתות בגרף בזו אחר זו על ידי הפעלת הכלל הכחול או הכלל האדום באופן שרירותי.
- עצור כאשר כל הקשתות צבועות - הקשתות הכחולות מגדירות עץ פורש מינימלי (עפ"מ)

הערה 3.3 קשת לעולם לא מחליפה צבע, הכלל הכחול והאדום מופעלים רק על קשתות לא צבועות.

3.1.1 הוכחת נכונות

טענה 3.4 תהא $e=(u,v)$ קשת בגרף ולא בעץ. אזי בתת הגרף $T+\{e\}$ יש מעגל יחיד שעובר דרך הקשת e. ניקח $T'=T+\{e\}-\{e'\}$ כאשר e' קשת בעץ במסלול (היחיד) מ-u ל-v, אזי T' מגדיר עץ פורש בגרף.

הוכחה: $|T|=n-1$ לכן - $|T'|=|T+\{e\}-\{e'\}|=n-1$.
בנוסף T' עדיין קשיר, כי השמטנו קשת e' ששייכת למעגל שמכיל את $e=(u,v)$.
לכן לפי משפט על עצים T' תת גרף קשיר עם $n-1$ קשתות הוא עץ. ■

משפט 3.5 בסיום האלגוריתם כל הקשתות צבועות והקשתות הכחולות מגדירות עץ פורש מינימום

הוכחה: נחלק את ההוכחה לשני חלקים -

1. נניח שבכל צעד ניתן להפעיל את הכלל הכחול או האדום, אזי בסיום הקשתות הכחולות מגדירות עץ פורש מינימום.
2. בכל שלב ניתן להפעיל את אחד הכללים.

אם נוכיח את שני השלבים, למעשה הוכחנו את המשפט.

1. נוכיח באינדוקציה שלאחר צביעת הקשתות $e_1,e_2,\ldots e_k$ יש בגרף עץ פורש מינימום שמכיל את כל הקשתות הכחולות מתוך $e_1,e_2,\ldots e_k$ ואף קשת אדומה מתוך $e_1,e_2,\ldots e_k$.

- בסיס - עבור $k=0$ אין אף קשת צבועה, ברור שקיים עץ פורש מינימום שמכיל את כל הקשתות הכחולות ואף קשת אדומה.

- צעד - נניח כי הטענה נכונה עבור $e_1, e_2, \ldots e_{k-1}$ ובצעד ה-k צבענו את הקשת e_k, נפריד למקרים -
  - אם e_k נצבע על פי הכלל הכחול. נניח בשלילה שהטענה לא נכונה, כלומר אין עפ"מ שמכיל את כל הקשתות הכחולות ואף קשת אדומה (מתוך הקשתות שצבענו). יהא T עפ"מ שמכיל את כל הקשתות הכחולות מתוך $e_1, e_2, \ldots e_{k-1}$ ואף קשת אדומה, מההנחה בשלילה ברור כי $e_k \notin T$.
  כדי לצבוע את e_k בכחול הפעלנו על חתך $(\bar{x}, x)$ את הכלל הכחול. ברור כי $e_k = (u, v)$ לא הייתה צבועה באדום, ולא בכחול. בעץ T יש מסלול מ-u ל-v (עץ, לכן קשיר) ולכן חייבת להיות בו קשת חתך אחרת e עבור החתך $(\bar{x}, x)$ הנ"ל. נחליף את e ב-e_k ב-T, וקיבלנו עץ פורש חדש שמכיל את כל הקשתות הכחולות ואף קשת אדומה, לפי הכלל הכחול $W(e_k) \leq W(e)$ ולכן
  $$w(T + \{e_k\} - \{e\}) \leq w(T)$$
  כלומר העץ החדש גם הוא עפ"מ - סתירה.
  - אם e_k נצבע על פי הכלל האדום - נניח בשלילה כי כל עץ פורש מינימום שמכיל את כל הכחולות ואף אדומה מתוך $e_1, e_2, \ldots e_{k-1}$ חייב להכיל את e_k.
  יהא T עפ"מ שנבנה על פי הכלל עבור $e_1, e_2, \ldots e_{k-1}$, מההנחה אם נוציא מ-T את e_k הוא יתפרק לשני תת עצים - T_1 ו-T_2. נתבונן על המעגל שבגללו צבענו את e_k באדום - במעגל זה חייבת להיות לפחות קשת אחת e שמחברת בין T_1 ו-T_2. כיוון שצבענו את e_k באדום נקבל כי $W(e_k) \geq W(e)$, נתבונן שוב על העץ הפורש שנוצר ע"י - $T' = T + \{e\} - \{e'\}$ ובדיוק באותו האופן - $w(T + \{e_k\} - \{e\}) \leq w(T)$ נקבל כי T' עפ"מ שאינו מכיל את e_k - סתירה.

2. [5]מדוע תמיד ניתן להפעיל את אחד הכללים?

הקשתות הכחולות תמיד מגדירות יער - כי יש עפ"מ שמכיל את כולן. נתבונן על קשת שלא צבועה -

(א) אם היא סוגרת מעגל באחד העצים ביער - היא הקשת היחידה שאינה צבועה במעגל זה ולכן ניתן לצבוע אותה באדום.

(ב) אם הקשת e מחברת שני עצים $(T_1,\ T_2)$ ביער - נחלק את העצים ביער לשתי קבוצות, כך ש-T_1 נמצא בקבוצה אחת ו-T_2 בשניה. כיוון שכל עץ נמצא בקבוצה נפרדת הרי שבחתך אין אף קשת כחולה, ויש לפחות קשת אחת בחתך (הקשת e) לכן ניתן להפעיל את הכלל הכחול.

■

3.2 תנאים לאופטימליות

בהנתן עץ T, אילו תנאים ניתן לבדוק כדי לוודא כי T אכן עפ"מ?

נשים לב כי כל קשת e בעץ מגדירה חתך - אם נסיר אותה מהעץ נקבל שני רכיבים, נסמן חתך זה ב-c_e.

טענה 3.6 אם לכל קשת $e \in T$ בחתך c_e לקשת e יש את המשקל המינימלי - אזי T עפ"מ.

הוכחה: ראשית נשים לב כי התנאי הכרחי. אם יש חתך c_e שבו e איננה הקשת המינימלית אזי ניתן לקבל עץ במשקל קטן יותר ע"י החלפת e בקשת הקלה יותר.

לכל חתך c_e שמוגדר ע"י הקשת $e \in T$ יש רק נציג אחד בעץ - הקשת e. נציע דרך לבנות את העץ T באמצעות אלגוריתם שמתאים לדרישות של האלגוריתם הכללי לבניית עפ"מ - וכך נראה כי T אכן עפ"מ. נדון בכל פעם בחתך

[5] הרצאה 5 16.11.09

c_e - אין בחתך הזה אף קשת כחולה, ו-e היא הקשת בעלת המשקל המינימלי בחתך - לכן האלגוריתם יצבע אותה בכחול. ∎

טענה 3.7 אם לכל קשת $e \notin T$ מתקיים כי e היא הקשת הכבדה ביותר במעגל ש-e סוגרת בעץ אזי T עפ"מ.

הוכחה: נפעיל את הכלל האדום על כל המעגלים מהצורה -

$$e = (u, v) + L$$

כאשר L הוא המסלול ב-T בין u, v, ו-$e \notin T$. הקשת e היא הכבדה ביותר במעגל ולכן נצבע אותה באדום (קשתות העץ אינן נצבעות באדום ולכן אין אף קשת אדומה). בסיום התהליך כל הקשתות מחוץ ל-T צבועות באדום - כעת אין מנוס מלהפעיל את הכלל הכחול עד שכל שאר הקשתות יצבעו בכחול - ונקבל עפ"מ. ∎

3.3 אלגוריתם Kruskal

- מיין את כל הקשתות בגרף לפי משקלן מקל לכבד.
- עבור על הקשתות לפי סדר המשקל, ולכל קשת e בתורה -
  - אם e מחברת בין שני צמתים ששייכים לאותו עץ כחול - צבע את e באדום.
  - אחרת - צבע את e בכחול
- עצור כאשר הקשתות הכחולות מגדירות עץ פורש (כאשר נצבעו $|V| - 1$ קשתות בכחול)

בכל שלב האלגוריתם מחזיק יער של עצים כחולים. בהתחלה כל עץ ביער הוא צומת בודד.

מדוע האלגוריתם מוצא עפ"מ? על כל קשת e שנבדקת (ואינה סוגרת מעגל כחול) ניתן להפעיל את הכלל הכחול כי בחתך שבין T_1 ל-T_2 (+הרכיבים האחרים) אין אף קשת כחולה - כיוון שהאלגוריתם בוחר את הקשתות מקלה לכבדה - ברור ש-e הקשת הקלה ביותר בחתך, אחרת היינו בוחנים קשת אחרת מהחתך קודם לכן.

3.3.1 סיבוכיות

- מיון - $O\left(|E|\, log\left(|E|\right)\right) = O\left(|E|\, log\left(|V|\right)\right)$.
- נחזיק $union - find$ של העצים ביער, עבור כל צומת נבדוק האם היא מחברת שתי קבוצות שונות (נצבע בכחול) או נמצאת בתוך אותה קבוצה (נצבע באדום) סה"כ סיבוכיות - $O\left(|E| \cdot log^{*}\left(|V|\right)\right)$.

3.4 האלגוריתם של Prim

- אתחול - $R = \{v_1\}$ (R היא הקבוצה שכבר נפרשה ע"י האלגוריתם)
- הפעל את הכלל הכחול על החתך $R, V \backslash R$, ובחר את הקשת המינימלית $e = (u, v)$ (כך ש-$u \in R, \quad v \notin R$) והוסף את v ל-R (כלומר - $R = R \cup \{v\}$)
- כל עוד $R \neq V$ חזור על השלב הקודם.

נכונות האלגוריתם - מתקבלת מיידית. בכל צעד מפעילים את הכלל הכחול על חתך שאין בו קשת כחולה.

3.4.1 סיבוכיות

יש $|V|-1$ צעדים שבהם מפעילים את הכלל הכחול, ניתן לממש באמצעות ערימה ולקבל סיכוביות כוללת $O\left(|E|\,log\left(|V|\right)\right)$.

4 מסלולים קלים ביותר[6]

יהא $G=(V,E)$ גרף מכוון, ו-$w:E\to\mathbb{R}$ פונקציית משקל על הקשתות. נתבונן על מסלול -

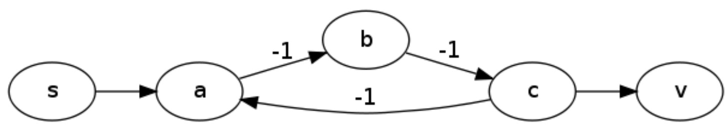
$$p=v_1\overset{e_1}{\to}v_2\to\cdots\to v_{n-1}\overset{e_{n-1}}{\to}v_n$$

ונגדיר משקל למסלול - $w\left(p\right)=\sum_{i=1}^{n-1}w\left(e_i\right)$.

בהנתן צומת מקור s נרצה למצא את המסלול הקל ביותר בין s לצומת כלשהו v, ונסמן -

$$\delta\left(v\right)=\delta\left(s,v\right)=\min_{s\overset{p}{\rightsquigarrow}v}\left\{w\left(p\right)\right\}$$

נשים לב שגם אם קיים מסלול מ-s ל-v המסלול הקל ביותר לא בהכרח מוגדר, למשל -

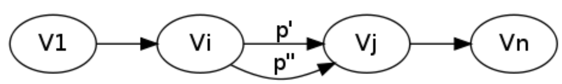


איור 5: מסלול שבו יש מעגל שלילי - המסלול הקל ביותר הוא במשקל מינוס אינסוף - למעשה לא מוגדר

לכן מעתה נניח כי ייתכנו משקלי קשתות שליליים, אבל אין מעגלים שליליים.

טענה 4.1 בהנתן מסלול קל ביותר $p:v_1\rightsquigarrow v_n$ גם כל תת מסלול שלו $p':v_i\rightsquigarrow v_j\subseteq p$ (לכל $i\le j$) הוא מסלול קל ביותר מ-v_i ל-v_j.

הוכחה: אם תת המסלול מ-v_i ל-v_j לא היה הקל ביותר ניתן היה למצא מסלול קל יותר מ-v_1 ל-v_n. כלומר -



איור 6: אם $w\left(p''\right)<w\left(p'\right)$ הרי שניתן לבנות מסלול p^* חדש דרכו קצר יותר מ-p

בסתירה לנתון כי p מסלול קצר ביותר ■

טענה 4.2 ("אי שוויון המשולש") לכל מקור s מתקיים - $\delta\left(s,v\right)\le\delta\left(s,u\right)+w\left(\left(u,v\right)\right)$

הוכחה: $\delta\left(s,v\right)$ חסום מלמעלה על ידי משקל כל מסלול מ-s ל-u (בפרט המסלול הקל ביותר מ-s ל-u) + הקשת $\left(u,v\right)$ ■

[6] הרצאה 6 23.11.09

4.1 אלגוריתם Dijkstra

נניח שבגרף שלנו אין משקלים שליליים, כלומר - $w: E \to \mathbb{R}^{+,0}$, במקרה זה הבעיה מעט יותר פשוטה, ונוכל לפתח את אלגוריתם דייקסטרה (Dijkstra).

<u>המטרה:</u> בהנתן צומת s לחשב את $\delta(s,v)$ לכל צומת $v \in V$

4.1.1 האלגוריתם

- לכל צומת $s \neq v$ אתחל $d(v) = \infty$.
- אתחל $d(s) = 0$
- $V \to T$
- כל עוד $T \neq \emptyset$ בצע
  - יהא u הצומת בעל $d(u)$ מינימלי מתוך כל $u \in T$.
  - $T = T/\{u\}$ (הסר את u מ-T)
  - לכל אחד מהשכנים[7] של u שנמצאים עדיין ב-T (כלומר - $v \in N(u) \cap T$) עדכן -

$$d(v) \leftarrow min\{d(v), d(u) + w(u,v)\}$$

    * אם אכן עדכנו את $d(v)$ עדכן גם $\pi(v) = u$.

4.1.2 הוכחת נכונות

טענה 4.3 אם $d(v)$ סופי, אזי יש מסלול מ-s ל-v סופי שמשקלו $d(v)$. המסלול מתקבל מתוך המצביעים π (על ידי הליכה "אחורה").

הוכחה: באינדוקציה על סדר היציאה מהקבוצה T

- בסיס - הצומת הראשון שיצא מ-T הוא s ועבורו ידוע לנו שיש מסלול ריק (s בלבד) מ-s לעצמו.
- צעד - נניח שכל הצמתים שיצאו מ-T לפני v מקיימים את הטענה. $d(v)$ סופי, ולכן קיים צומת u שעדכן את $d(v)$ אחרון (כלומר $\pi(v) = u$) וכן מתקיים - $d(v) = d(u) + w((u,v))$. לפי הנחת האינדוקציה קיים מסלול מ-s ל-u שמשקלו $d(u)$, נשרשר לסוף המסלול הזה את (u,v) וקיבלנו מסלול במשקל $d(v)$ מ-s ל-v. המסלול מתקבל על פי המצביעים π, נתחיל מ-v, נעבור ל-$u = \pi(v)$ ונמשיך לאורך המסלול הקיים על פי הנחת האינדוקציה.

■

טענה 4.4 לכל צומת v, אלגוריתם דייקסטרה מחשב את המסלול הקל ביותר כלומר בסיום $\delta(v) = d(v)$.

הוכחה: באינדוקציה על סדר יציאת הצמתים מ-T

[7] בגרף מכוון, הצמתים שיש קשת מהם ל-u

- **בסיס** - s הצומת הראשון שיוצא מ-T ועבורו $\delta\left(s\right)=0$ וגם $d\left(s\right)=0$.
- **צעד** - נניח באינדוקציה עבור כל הצמתים שיצאו עד עכשיו מ-T ונניח כי v הצומת שנבחר באינטרציה הנוכחית (כלומר הצומת בעל $d\left(v\right)$ המינימלי), עבור $d\left(v\right)<\infty$ (אחרת אין מה להוכיח - הצמתים כלל לא ישיגים מ-s).

נתבונן על המסלול הקצר ביותר מ-s ל-v ונסמן את הקשת האחרונה במסלול זה שחוצה את החתך $\{T,V\backslash T\}$ ב-$\left(x\to y\right)$ כלומר $x\in V\backslash T$ ו-$y\in T$. מהנחת האינדוקציה - $d\left(x\right)=\delta\left(x\right)$. כיוון שהמסלול $s\rightsquigarrow y$ הוא תת מסלול של מסלול קל ביותר ($s\rightsquigarrow v$) מתקיים כי $\delta\left(y\right)=\delta\left(x\right)+w\left(x\to y\right)$ ולכן גם - $\delta\left(y\right)=d\left(x\right)+w\left(x\to y\right)$.

אבל כאשר x יצא מ-T הוא עדכן את כל שכניו, ובפרט את y, כלומר $d\left(y\right)=min\left\{ d\left(y\right),d\left(x\right)+w\left(x\to y\right)\right\}$, מהטענה הקודמת מתקיים תמיד - $d\left(y\right)\geq\delta\left(y\right)$ ולכן כאשר x יצא מ-T הוא עדכן את $d\left(y\right)$ להיות

$$d\left(y\right)\;=\;d\left(x\right)+w\left(x\to y\right)=\delta\left(y\right)$$

נשים לב כי מתקיים גם $\delta\left(y\right)\leq\delta\left(v\right)$ (כי המשקלים אי שליליים ו-y מופיע לפני v על המסלול הקל ביותר מ-s ל-v) וקיבלנו -

$$d\left(y\right)=\delta\left(y\right)\leq\delta\left(v\right)\leq d\left(v\right)$$

אבל v האו בעל $d\left(v\right)$ המינימלי מתוך הצמתים ב-T, לכן - $d\left(v\right)\leq d\left(y\right)$ ולכן כל האי שוויוים למעלה הם למעשה שוויונים -

$$d\left(y\right)=\delta\left(v\right)=\delta\left(v\right)=d\left(v\right)$$

והוכחנו את טענת האינדוקציה.

■

4.1.3 סיבוכיות

בכל שלב צריך למצא את הצומת שיש לו תווית מינימלית ב-T.

- נאיבי - בסיבוכיות $\left|T\right|-1$ כלומר $O\left(\left|V\right|^{2}+\left|E\right|\right)$
- ע"י שימוש בערימה - $O\left(\left|E\right|log\left(\left|V\right|\right)\right)$
- ע"י שימוש בערימת פיבונאצ'י - $O\left(\left|E\right|+\left|V\right|log\left(\left|V\right|\right)\right)$
- אם כל המשקלים זהים (או סט סגור של משקלים עם יחס רציונלי בינהם) עדיף להשתמש ב-BFS ולקבל $O\left(\left|V\right|+\left|E\right|\right)$.

4.1.4 מה אם יש משקלים שליליים?

נגדיר פעולה שנקראת החלשה (Relexation) -

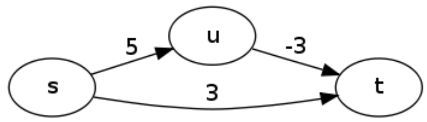
- נתבונן על קשת $u\to v$. אם $d\left(v\right)>d\left(u\right)+w\left(u\to v\right)$ אזי v מעדכן את התווית שלו להיות

$$d\left(v\right)\equiv d\left(u\right)+w\left(u\to v\right)$$

וכן מעדכן $\pi\left(v\right)\equiv u$.

[8]אם מתקיים התנאי לרלקסציה - יש בעיה בערך של $d\left(v\right)$, כלומר נרצה ליצור אלגוריתם רלקסציה - שיעדכן את התוצאות שמתקבלות ע"י דייקסטרה.

[8]הרצאה שביעית - 30.10.09



איור 7: דוגמא לגרף פשוט עם קשת שלילית, דייקסטרה במקרה זה ימצא את המסלול במשקל 3 מ-s ל-t, ויפספס את המסלול דרך u שהוא במשקל 2.

אתחול

- $d(s) \leftarrow 0$
- $d(v) \leftarrow \infty \quad \forall v \in V \backslash s$
- $\pi(v) \leftarrow NIL$

טענה 4.5 נניח שאתחלנו את d כפי שצויין, ונניח שביצענו פעולות רלקסציה כולל חזרות. בכל שלב מתקיים $d(v) \geq \delta(s,v)$ לכל $v \in V$.

הוכחה: נוכיח באינדוקציה

- הטענה נכונה במצב ההתחלתי - כיוון שאין מעגלים שליליים - $\delta(s,s) = 0 = d(s)$
- נניח בשלילה שהטענה אינה נכונה. יהא v הצומת הראשון שעבורו $d(v) < \delta(s,v)$ ותהא $u \rightarrow v$ הקשת שהרלקסציה שלה גרמה לטענה להיות לא נכונה.

  כאשר קרתה הרלקסציה $u \rightarrow v$ קבענו -

  $$d(u) + w(u \rightarrow v) = d(v)$$

  אבל לפי הנחת הראשוניות (של השלילה) $d(u) \geq \delta(u)$. מצד שני $d(v) < \delta(v)$ כלומר -

  $$\begin{aligned} d(u) + w(u \rightarrow v) &= d(v) < \delta(v) \leq \delta(s,u) + w(u \rightarrow v) \\ &\Downarrow \\ d(u) &< \delta(s,u) \end{aligned}$$

  וזו כמובן סתירה.

■

מסקנה 4.6 כאשר $d(v) = \delta(s,v)$ עבור צומת v הערך של $d(v)$ לא ישתנה יותר.

4.2 אלגוריתם בלמן-פורד Bellman-Ford

(אלגוריתם למציאת מסלולים קלים ביותר לכל צומת משורש נתון - גם במקרה שבו יש קשתות שליליות)

- אתחול -

- $d(s)=0$
- $d(v)=\infty \ \forall v\neq s$
- $\pi(v)=NIL \quad \forall v$

- לולאה, בצע $|V|-1$ פעמים -
  - עבור על כל הקשתות $(\forall e: u\to v)$ -
    - אם $d(v)>d(u)+w(e)$ בצע רלקסציה -
      - $d(v)=d(u)+w(e)$
      - $\pi(v)=u$

- אם בסיום האלגוריתם קיימת קשת $e: u\to v$ שעבורה לא מתקיים אי שוויון המשולש - הודע שקיים מעגל שלילי.

הערה 4.7 אפשר לעצור את האלגוריתם אחרי איטרציה של הלולאה הראשית שבה לא התבצעה אף פעולת רלקסציה.

4.2.1 מדוע האלגוריתם מחשב את המסלול הקל ביותר? (הוכחת נכונות)

משפט 4.8 נניח כי המסלול הקל ביותר מ-s ל-v מכיל k קשתות. אזי לכל המאוחר בסיום האיטרציה ה-k של אלגוריתם בלמן פורד $d(v)=\delta(s,v)$.

הוכחה: באינדוקציה על k -

- כאשר $k=0$ בהכרח $s=v$ וכבר באתחול $d(s)=0=\delta(s,s)$.
- צעד - נניח כי המשפט נכון עבור $k-1$ ונוכיח עבור k. נתבונן על מסלול קל ביותר מ-s ל-v -

$$s\to v_1\to v_2\rightsquigarrow\cdots\to v_{k-2}\to v_{k-1}\to v_k=v$$

המסלול - $s\rightsquigarrow v_{k-1}$ הוא מסלול קל ביותר ל-v_{k-1}, ולכן לפי הנחת האינדוקציה הרי שבסיום האיטרציה ה-$k-1$ מתקיים $d(v_{k-1})=\delta(s,v_{k-2})$. באיטרציה ה-$k$ עוברים על כל הקשתות ובודקים אם אפשר לבצע רלקסציה, בפרט - בודקים אם $d(v_k)>d(v_{k-1})+w((v_{k-1}\to v_k))$, אם כן מעדכנים -

$$d(v)=d(v_k)=d(v_{k-1})+w((v_{k-1}\to v_k))=\delta(s,v_k)$$

נשים לב שלא יתכן ש-

$$d(v_k)<d(v_{k-1})+w((v_{k-1}\to v_k))$$

מפני שאז $d(v_k)<\delta(s,v_k)$ והוכחנו שזה לא יתכן.

■

4.2.2 סיבוכיות

בכל איטרציה בודקים כל קשת - $O(|E|)$. יש לכל היותר $|V|-1$ איטרציות, לכן סיבוכיות - $O(|E|\cdot|V|)$.

4.2.3 עץ מסלולים קלים ביותר

כאשר האלגוריתם מסיים את הריצה המצביעים π מגדירים עץ מסלולים קלים ביותר.

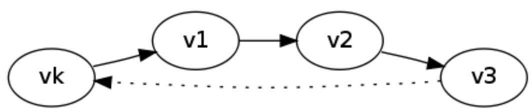
1. בכל שלב, אם $d(v) < \infty$ אזי המצביעים π מגדירים מסלול במשקל $d(v)$ מ-s ל-v. מדוע? באינדוקציה על מספר פעולות הרלקסציה.

- כאשר עושים רלקסציה לקשת $u \rightarrow v$ ומעדכנים את $d(v)$ לפי הנחת האינדוקציה יש מסלול שמשקלו $d(u)$ עד לצומת u. לאחר הרלקסציה $d(v) = d(u) + w((u \rightarrow v))$ ו-$\pi(v) = u$ לכן הנחת האינדוקציה נשמרת.

2. הגרף שמוגדר ע"י המצביעים π חייב להיות עץ מכוון. אם הוא מכיל מעגלים, אזי אם המעגל מכוון - הוא לא יכול להכיל את s (כי $\pi(s)$ לא מעודכן לעולם), אחרת יש במעגל צומת עם דרגת כניסה 2 - וזו סתירה (כי לכל צומת יש "אב" יחיד המוגדר ע"י π). אם המעגל לא מכוון אזי יש צומת שדרגת הכניסה שלו היא 2 - וזו שוב סתירה.

4.2.4 מה קורה אם יש מעגל שלילי?

נראה שבמקרה כזה אף פעם לא נגיע למצב שבו אי אפשר לעשות רלקסציות -



איור 8: נניח מעגל באורך k כך ש-$\sum_{i=1}^{k} w(v_i \rightarrow v_{i+1}) < 0$

נניח בשלילה כי יש סדרת תווית $d(v_i)\ \ \forall i \in [1,k]$ שמקיימת את אי שוויון המשלוש -

$$\begin{aligned} d(v_2) &\leq d(v_1) + w(v_1 \rightarrow v_2) \\ d(v_3) &\leq d(v_2) + w(v_2 \rightarrow v_3) \\ &\vdots \\ d(v_1) &\leq d(v_k) + w(v_k \rightarrow v_1) \end{aligned}$$

כלומר -

$$\begin{aligned} \sum_{i=1}^{k} d(v_i) &\leq \sum_{i=1}^{k} d(v_i) + \sum_{i=1}^{k} w(v_i \rightarrow v_{i+1}) \\ 0 &\leq \sum_{i=1}^{k} w(v_i \rightarrow v_{i+1}) \end{aligned}$$

כלומר סכום הקשתות במעגל אי שלילי - בסתירה לנתון.

4.3 תכנון דינמי

נסמן $d_{ij}^{(k)}$ משקל המסלול הקל ביותר מ-i ל-j כאשר הצמתים במסלול שייכים לקבוצה - $(1,2,\dots k)$. לדוגמא -

$$d_{ij}^{(0)} = \begin{cases} w\left(i\rightarrow j\right) & \text{if exists} \\ \infty & \text{Otherwise} \end{cases}$$

בתכנון דינמי נרצה לפתור בעיות על ידי פתרון בעיות קטנות (קלות) יותר. נניח כי אנחנו יודעים את $d_{ij}^{(k-1)}$ איך ניתן לחשב את $d_{ik}^{(k)}$? יש שתי אפשרויות -

- אם המסלול הקצר ביותר עם הצמתים $(1,2,\dots k)$ כלל לא כולל את k נקבל כי $d_{ij}^{(k)} = d_{ij}^{(k-1)}$.
- אחרת - $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$

4.3.1 אלגוריתם לחישוב המסלולים הקלים ביותר לכל זוג צמתים i,j

- אתחול -
  - $\forall i,j:\ d_{ij}^{(0)} = w\left(i\rightarrow j\right)$ אם קיימת קשת כזו, אחרת $d_{ij}^{(0)} = \infty$
- לכל $k=1$ עד $|V|$ בצע -
  - לכל $i=1$ עד $|V|$ בצע -
    - לכל $j=1$ עד $|V|$ בצע -
      - $d_{ij}^{(k)} = min\left\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$

סיבוכיות יש שלוש לולאות מקוננות, בכל לולאה רצים מ-1 ועד $|V|$ לכן - $O\left(|V|^3\right)$.

4.3.2 בניה אחרת

נגדיר $d_{ij}^{(m)}$ להיות המסלול הקל ביותר מ-i ל-j שמכיל לכל היותר m קשתות. לדוגמא -

$$d_{ij}^{(1)} = \begin{cases} w\left(i\rightarrow j\right) & \text{if exists} \\ \infty & \text{Otherwise} \end{cases}$$

במקרה זה יתקיים -

$$d_{ij}^{(m)} = \min_{v\in V}\left\{d_{iv}^{(m-1)} + w\left(v\rightarrow j\right)\right\}$$

כאן אנחנו למעשה מחפשים את המסלול הקל ביותר שאורכו לכל היותר $m-1$ לכל צומת אחר - שאליו משורשרת קשת אל j - כלומר מוצאים את המסלול הקל ביותר באורך שהוא לכל היותר m בין i ו-j.
האלגוריתם - טרוויאלי, והסיבוכיות שתתקבל - $O\left(|V|^4\right)$.

5 אלגוריתמים חמדניים[9]

נרצה לפתור בעיית אופטימיזציה, למשל - בעיה בגרפים. בפתרון הבעיה מבצעים פעולה איטרטיבית - ובכל צעד מבצעים פעולת חישוב. אלגוריתם חמדן יבצע בכל צעד את הפעולה ה"זולה" ביותר. דוגמא טובה לאלגוריתם חמדן היא האלגוריתמים שראינו לבניית עפ"מ - בכל צעד הסתכלנו על חתך ובחרנו את הקשת הקלה ביותר, או שהסתכלנו על מעגל וזרקנו את הקשת הכבדה ביותר.

בדרך כלל בבעיות אופטימיזציה אלגוריתם חמדן לא מסוגל לתת את הפתרון הטוב ביותר, כלומר - כדי להגיע לפתרון האופטימלי בדרך כלל יש צורך לעשות "בדרך" צעדים לא אופטימליים.

5.1 שיבוץ משימות

נתונות n משימות, נסמן $J_1, J_2, \dots J_n$ כל משימה מורכבת מאינטרוול בציר הזמן - שמציין את ההתחלה והסיום שלה. זה למשל יכול למדל מכונה, שצריך לשבץ את המשימות שהיא תבצע תחת האילוצים הבאים -

- המכונה יכולה לבצע רק משימה אחת בכל זמן נתון.
- לא ניתן לבצע משימה באופן חלקי (חייבים להתחיל מזמן ההתחלה ולסיים בסוף).

המטרה מציאת שיבוץ חוקי שממקסם את מספר המשימות שמשובצות.

למה 5.1 המשימה J שנקודת הסיום שלה היא השמאלית ביותר מבין כל האינטרוולים (המשימה שמסתיימת ראשונה) שייכת לפתרון אופטימלי.

הוכחה: נניח כי J לא שייכת לפתרון אופטימלי כלשהו OPT, כלומר בהכרח בפתרון האופטימלי יש משימה נוספת J' שנחתכת עם J. מהגדרת J מתקיים בהכרח ש-J' מסתיים אחרי (או באותו זמן כמו) J.

נתבונן על הפתרון האופטימלי ללא J' ובתוספת J, לא ייתכן ש-J מתנגש עם משימה שמתחילה אחרי J' (כי J' מסתיים אחרי או באותו זמן כמו J). לא ייתכן שיש ב-OPT משימה שמתחילה לפני J בפתרון (כי היא תסתיים אחרי תחילת J' ותתנגש עם J'). כלומר - זה הוא שיבוץ חוקי והוא מכיל אותו מספר משימות כמו הפתרון האופטימלי - לכן אופטימלי. ■

5.1.1 אלגוריתם חמדן -

- מיין את האינטרוולים לפי נקודת הסיום, הסדר הממויין יהיה -

  $J_1, J_2, \dots J_n$

- כל עוד יש אינטרוולים -
  - הכנס את האינטרוול הראשון (J) בסדר הממויין לפתרון
  - זרוק את כל האינטרוולים שנחתכים עם J

[9] הרצאה שמינית - 7.12.09

נכונות   האלגוריתם מחזיר פתרון אופטימלי כיוון שיש פתרון אופטימלי שמכיל את המשימה שמסתיימת ראשונה, המשך באינדוקציה.

5.2 שיבוץ משימות - בעיה מורכבת יותר

נתונות n משימות, נסמן $J_1, J_2, \dots J_n$. ולכל משימה J_i נתון זמן ריצה t_i וזמן סיום רצוי d_i.

- בכל רגע נתון המכונה יכולה לבצע רק משימה אחת.
- כל משימה צריכה להתבצע ברצף.

כלומר לכל משימה צריך לקבוע זמן התחלה s_i ואז זמן הסיום נתון ע"י $f_i = s_i + t_i$.

הגדרה 5.2 איחור של משימה i הוא $l_i = \max\{0, f_i - d_i\}$.

המטרה   מצא שיבוץ חוקי של המשימות שממזער את האיחור המקסימלי -

$$\min_{\substack{valid \\ arrangements}} \left\{ \max_{i \le n} \{l_i\} \right\}$$

דוגמא לכללים חמדניים לא מוצלחים

- שבץ את המשימות לפי t_i כדי להפטר ממשימות קצרות. דוגמא נגדית -

$$t_1 = 1 \qquad d_1 = 100$$
$$t_2 = 10 \qquad d_2 = 10$$

- שבץ את המשימות לפי $d_i - t_i$. דוגמא נגדית -

$$t_1 = 1 \qquad d_1 = 2$$
$$t_2 = 10 \qquad d_2 = 10$$

האלגוריתם **(EDF - Earliest Deadline First)**

- נמיין את המשימות ע"פ זמן הסיום שלהן -

$$d_1 \le d_2 \le \dots \le d_n$$

- אתחול $f = 0$.
- עבור על המשימות לפי זמן הסיום הממויין, לכל i -
  - $s_i = f$
  - $f_i = s_i + t_i$
  - $f = f_i$

הוכחת נכונות

טענה 5.3 קיים פתרון אופטימלי שאין בו אין זמני סרק (זמנים שהמכונה לא עובדת בהם)

הוכחה: אם יש זמן סרק נקדים את המשימה שאחריו כך שתתבצע מוקדם יותר. ■

הגדרה 5.4 לשיבוץ יש החלפה אם יש משימה i עם זמן סיום d_i שמשובצת לפני משימה j עם זמן סיום d_j כך ש-$d_j < d_i$.

טענה 5.5 לכל השיבוצים שאין להם זמני סרק ואין בהם החלפות יש אותו איחור.

הוכחה: בהנתן שני שיבוצים בלי זמני סרק ובלי החלפות ההבדל היחיד שייתכן הוא שתי משימות עם אותו deadline שהסדר בינהן הוחלף. מבין המשימות האלו המשימה בעלת האיחור המקסימלי היא האחרונה, אבל זמן הסיום שלה לא תלוי בסידור הפנימי בין המשימות הללו. ■

כדי להוכיח את אופטימליות אלגוריתם EDF נותר להוכיח שיש שיבוץ אופטימלי שאין בו החלפות - כיוון ש-EDF מייצר כזה שיבוץ הרי ש-EDF אופטימלי.

נתחיל עם שיבוץ אופטימלי שיש בו החלפות. אם יש זוג משימות שיש להן החלפה אז קיימות שתי משימות עוקבות שיש להן החלפה. נבצע החלפה בין שתי המשימות העוקבות הללו. מה יקרה כתוצאה מההחלפה?

- היא משפיעה רק על האיחורים של a, b.
- האיחור של b (המשימה שהייתה מאוחרת יותר) רק השתפר (קטן).
- האיחור של a -

$$l\left(a\right) = f_a - d_a < \underbrace{f_b - d_b}_{\substack{before \\ \text{substitute}}}$$

לכן מקסימום האיחורים לא גדל, כלומר השיבוץ נותר אופטימלי. ניתן להמשיך ולבצע החלפות באותות האופן עד שלא ניתן לבצע החלפות.

5.3 עצי Huffman[10]

נתון קובץ המורכב מתווים. לכל תו נתונה הסתברות מופע בקובץ. נרצה לקודד את הקובץ באופן בינארי כאשר המטרה היא למזער את אורך הקובץ המקודד.

5.3.1 קידוד

- מילת קוד מעל $\{0,1\}$ היא - $w = a_1 a_2 \ldots a_n$ כאשר $a_i \in \{0,1\}$.
- אורך המילה יסומן - $l\left(w\right)$
- אוסף של מילות קוד לא ריקות יקרא קוד.

[10] הרצאה תשיעית 14.12.09

דוגמא -

$$C = \{c_1, c_2, c_3\}$$
$$c_1 = 01 \quad c_2 = 0 \quad c_3 = 10$$

קוד יקרא חד פענח אם יש רק אפשרות אחת לפענח אותו. בדוגמא C אינו חד פענח, למשל -

$$010 = \begin{cases} 01, 0 & c_1 c_2 \\ 0, 10 & c_2 c_3 \end{cases}$$

קוד ייקרא חסר רישאות אם אין אף מילת קוד שהיא רישא של מילה אחרת. במקרה כזה הפענוח ייעשה על ידי קריאה משמאל לימין וזיהוי מילת קוד ברגע שמזהים אחת (קוד חסר רישאות הוא קוד חד פענח). קוד חסר רישאות ניתן לייצג בעזרת עץ - כך שמילות קוד מתאימות לעלים.

הגדרת הבעיה

נתונים n תווים, לכל תו i יש הסתברות של $f(i)$ להופיע בקובץ (או, מופיע בקובץ $f(i)$ פעמים) רוצים למצא קידוד שממזער את אורך הקובץ, כלומר אם c_i הוא אורך מילת הקוד המתאימה לתו i, נחפש קידוד שממזער את -

$$\sum_{i=1}^{n} f(i) c_i$$

טענה 5.6 (ללא הוכחה) קיים קוד אופטימלי לבעיה שהוא קוד חסר רישאות (עומק העלה בעץ שמייצג את הקוד הוא אורך מילת הקוד)

המטרה - קיים T - עץ המתאים לקוד חסר רישאות כך ש-

$$cost(T) = \sum_{i=1}^{n} f(i) d_T(i)$$

(כאשר - $d_T(i)$ עומק העלה שמייצג את התו ה-i בעץ T) עלינו למצא עץ בינארי T עם n עלים כך ש-$cost(T)$ מינימלי.

אינטואיציה - נרצה שעלה שמופיע בטקסט הרבה פעמים יופיע גבוה בעץ (מילת קוד קצרה), ועלה שמופיע בטקסט מעט פעמים יופיע נמוך בעץ (מילת קוד ארוכה).

5.4 אלגוריתם האפמן

1. כל עוד מספר מילות הקוד גדול מאחד -
   (א) קח את שתי מילות הקוד עם הסתברות המופע הנמוכה ביותר - x, y
   (ב) אחד את x, y למילת קוד חדשה z עם הסתברות מופעים -
   $$f(z) \equiv f(x) + f(y)$$
2. מילת הקוד האחרונה שנותרה היא שורש העץ, כעת נפצל כל "איחוד" שביצענו בשלב הקודם. כל מילת קוד שנוצרה כאיחוד של שתי מילות קוד תהיה צומת פנימי בעץ וכל מילת קוד מקורית תהפוך לעלה.

5.4.1 דוגמת הרצה

a	**b**	**c**	**d**	**e**	**f**
45	13	12	16	9	5

$\Downarrow$

a	**b**	**c**	**d**	**e, f**
45	13	12	16	14

$\Downarrow$

a	**b, c**	**d**	**e, f**
45	25	16	14

$\Downarrow$

a	**b, c**	**d, e, f**
45	25	30

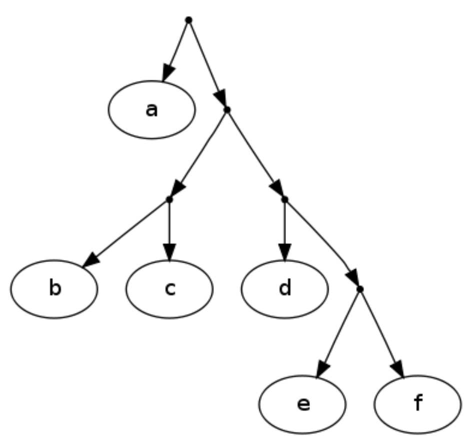
$\Downarrow$

a	**b, c, d, e, f**
45	55

$\Downarrow$

a, b, c, d, e, f
100

והעץ -



5.4.2 הוכחת אופטימליות אלגוריתם האפמן

טענה 5.7 עץ אופטימלי T הוא עץ שלם (כלומר - לכל צומת פנימי יש 2 בנים)

הוכחה: נניח בשלילה כי קיים עץ T אופטימלי ובו צומת פנימי u בעל בן אחד v. ניצור עץ T' חדש בו u לא קיים (האב של u מחובר ישירות ל-v), בעץ החדש כל עומק כל העלים שהם צאצאים של u ירד באחד, ולכן בהכרח

$$cost\left(T'\right) < cost\left(T\right)$$

בסתירה לאופטימליות. ■

טענה 5.8 יהיו x,y שתי מילים בעלות הסתברות המופעים הקטנה ביותר, קיים עץ אופטימלי T שבו x,y עלים אחים עמוקים ביותר בעץ (כלומר - הקידוד של x,y זהה עד כדי הביט האחרון)

הוכחה: נניח בשלילה שלא קיים T כזה. לפי טענה 5.7 קיימים בעץ זוג עלים אחים (עמוקים ביותר) b,c ונניח בה"כ - $f(b) \le f(c)$ ו-$f(x) \le f(y)$ אזי -

$$\begin{aligned} f(x) &\le f(b) \\ f(y) &\le f(c) \end{aligned}$$

ואם נחליף את $x \leftrightarrow b$ ו-$c \leftrightarrow y$ נקבל עץ חדש T' כך ש-

$$cost(T') \le cost(T)$$

והעץ T' מקיים את הדרישה. ■

מה קורה לעלות העץ באלגוריתם כשפותחים את העלה z לעלים x,y?

T' הוא העץ לפני פתיחת z

T העץ המתקבל אחרי הפתיחה.

$$\begin{aligned} cost(T) &= cost(T') - f(z)\, d_T(z) + [f(x) + f(y)]\,(d_T(z) + 1) = \\ &= cost(T') + f(x) + f(y) \end{aligned}$$

משפט 5.9 יהיו x ו-y שתי מילים עם תדירות המופעים הקטנה ביותר, $C' = C \setminus \{x,y\} \cup \{z\}$, ו-$f(z) = f(x) + f(y)$. אם T' הוא עץ אופטימלי ל-C' אז העץ T המתקבל מ-T' על ידי פתיחת העלה z לשני עלים x,y הוא אופטימלי ביחס ל-C.

הוכחה: נניח בשלילה כי קיים עץ T'' שהוא האופטימלי ביחס ל-C ומקיים -

$$cost(T'') < cost(T)$$

לפי טענה 5.8, נניח בה"כ ש-x,y עלים אחים נמוכים ביותר ב-T''.

נבנה עץ T''' באופן הבא - ניקח את T'' ונאחד את העלים האחים x,y לעלה אחד z, נשים לב ש-T''' הוא עץ לקוד C', ומתקיים -

$$cost(T''') = cost(T'') - f(x) - f(y) < cost(T) - f(x) - f(y) = cost(T')$$

כאשר סימן ה-$<$ מופיע מטענת השלילה. כלומר קיבלנו עץ T''' ל-C' שהעלות שלו קטנה יותר מהעלות של העץ האופטימלי T' - וזו סתירה! ■

6 תכנון דינמי

תכנון דינמי היא טכניקה המאפשרת פתרון של בעיה גם אם המרחב המעניין גדול מאד.

6.1 כפל מטריצות אופטימלי

נתון -

$$A_1 \cdot A_2 \cdots A_n$$

מטריצות, כאשר מטריצה A_i היא בגודל $p_{i-1} \times p_i$.

שאלה באיזה סדר כדאי להכפיל את המטריצות בשביל למזער את מספר הכפלים הסקלריים?

תזכורת כפל מטריצות הוא אסוציטיבי, ומספר הכפלים בפעולה $A_{p\times q}B_{q\times r}$ הוא $p \cdot q \cdot r$.

דוגמא $A_{10\times 100} \cdot B_{100\times 5} C_{5\times 50}$

- אופציה א -

$$(A \cdot B) \cdot C \Longrightarrow (10 \cdot 100 \cdot 5) + (10 \cdot 5 \cdot 50) = 7,500$$

- אופציה ב -

$$A \cdot (B \cdot C) \Longrightarrow (100 \cdot 5 \cdot 50) + (10 \cdot 100 \cdot 50) = 75,000$$

בהנחה שעלות של כפל סקלרי הוא קבוע, או שאין מידע מיוחד על מבנה המטריצות - ברור שאופציה א' עדיפה.
מה נעשה במקרה הכללי?

כמה אפשרויות יש לשים סוגריים?

$$P(n) = \begin{cases} 1 & n = 1 \\ \sum_{k=1}^{n-1} P(k)\, P(n-k) & n \neq 1 \end{cases}$$

כלומר - מספר קטלן של $n-1$ -

$$P(n) \quad = \quad C(n-1) = \frac{\binom{2n}{n}}{n+1}$$

עבור ערכי n גדולים מספר האפשרויות עצום.

6.1.1 מה הוא מספר המכפלות הסקלריות המינימלי על מנת לחשב מכפלה נתונה?

$$A_1 \cdot A_2 \cdots A_n$$

נגדיר $m(i,j)$ מספר המכפלות המינימלי שדרוש כדי לחשב מכפלה של המטריצות

$$A_i \cdot A_{i+1} \cdots A_j$$

נקבל את הנוסחה הרקורסיבית -

$$m\left(i,j\right)=\min_{i<k<j}\left\{m\left(i,k\right)+m\left(k+1,j\right)+p_{i-1}p_kp_j\right\}$$

עם תנאי ההתחלה -

$$m\left(i,i\right)=0$$

נרצה לבחור אסטרטגיית חישוב שתבטיח שכאשר אנחנו מחשבים את $m\left(i,j\right)$ כל הערכים $m\left(i',j'\right)$ שהוא תלוי בהם כבר חושבו קודם לכן. אם נתבונן על הטבלה -

$_i\backslash^j$	1	2	3			n
1	0					
2		0				
3			0			
				0		
					0	
n						0

התאים ה"מעניינים" לחישוב הם אלו שמתחת לאלכסון הראשי, נשים לב שתא תלוי בתאים שמשמאלו ומתחתיו - לכן נבחר בסידור של אלכסונים המקבילים לאלכסון הראשי ($\searrow$) ונחשב מהתא השמאלי התחתון ובכל פעם נוסיף אלכסון בכיוון $\nearrow$. סידור זה יבטיח שנחשב כל תא ב-$O\left(n\right)$ (כל התוצאות הדרושות מוכנות, וצריך לקחת מינימום על n איברים לכל היותר) ולכן בסך הכל החישוב כולו יתבצע ב-$O\left(n^3\right)$.

6.2 שיבוץ אינטרוולים[11]

נתון אוסף אינטרוולים על הקו הישר. כל אינטרוול מאופיין ע"י -

- s_i זמן ההתחלה של האינטרוול ה-i
- f_i זמן הסיום של האינטרוול ה-i
- w_i הרווח משיבוץ האינטרוול ה-i

המטרה - למצא $S\subseteq\{1,2,\dots n\}$ המביאה למקסימום את $\sum_{i\in S}w_i$ תחת האילוץ שכל שני אינטרוולים ב-S לא נחתכים.

למה האלגוריתם החמדן לא עובד? דוגמא -

$$\xrightarrow{w=3}$$

$$\xrightarrow{w=1}\xrightarrow{w=1}$$

האלגוריתם החמדן ירצה להביא למקסימום את מספר המטלות שהושלמו - אבל השלמת שתי המשימות תיתן משקל $w=1+1=2$ בעוד בחירת המשימה העליונה בלבד תיתן $w=3$.

נניח שהאינטרוולים ממוספרים לפי זמני הסיום - $f_1\leq f_2\leq\dots\leq f_n$. בנוסף נגדיר -

[11] הרצאה עשירית 21.12.09

- $p(j)$ - האינדקס הכי גדול של אינטרוול שאינו נחתך עם j (כלומר, הערך הכי גדול כך ש-$f_{p(j)} \leq s_j$). אם אין אינטרוול כזה נגדיר $p(j) \equiv 0$.

טענה 6.1 יהיה S^* פתרון אופטימלי לבעיה.

1. אם $n \notin S^*$ אז S^* אופטימלי לאינטרוולים $\{1, 2, \ldots n-1\}$.
2. אם $n \in S^*$ אז $S^* \backslash \{n\}$ אופטימלי לאינטרוולים $\{1, 2, \ldots p(n)\}$.

הוכחה:

1. נניח בשלילה שלא -

   $\Leftarrow$ קיים S' שהוא אופטימלי ל-$\{1, 2, \ldots n-1\}$ וגם -

$$\sum_{i \in S'} w_i > \sum_{i \in S^*} w_i$$

   ברור כי S' פתרון פיזבילי ל-$\{1, 2, \ldots n\}$ לכן S^* לא אופטימלי - וזו סתירה.

$\blacksquare$

נסמן $OPT(i)$ - ערך הפתרון האופטימלי עברו האינטרוולים $\{1, 2, \ldots i\}$. לכן -

$$OPT(i) = max \left\{ \underbrace{OPT(i-1)}_{i \notin Optimal}, \underbrace{w_i + OPT(p(i))}_{\text{i in optimal solution}} \right\}$$

כלומר - ניתן לחשב את $OPT(n)$ ע"י מעבר משמאל לימין, בסבוכיות $O(n)$ (בהנתן $p(i)$ וקלט ממויין).

6.3 בעיית Sequence Alignment

נניח שאנו מחפשים במילון "$ocurrance$" - המילה לא קיימת, אבל המחשב ישאל אותנו האם התכוונו ל-"$occurrence$"?

נשים לב כי -

$$\begin{array}{cccccccccc} o & \square & c & u & r & r & a & n & c & e \\ o & c & c & u & r & r & e & n & c & e \\ & \uparrow & & & & & \uparrow & & & \end{array}$$

כלומר קיים רווח מיותר, והחלפה אחת. לעומת זאת -

$$\begin{array}{ccccccccccc} o & \square & c & u & r & r & a & \square & n & c & e \\ o & c & c & u & r & r & \square & e & n & c & e \\ & \uparrow & & & & & \uparrow & \uparrow & & & \end{array}$$

שלושה רווחים - אבל אין חוסר התאמות.

הגדרת המרחק

$$\begin{aligned} X &= x_1x_2\dots x_m \\ Y &= y_1y_2\dots y_n \end{aligned}$$

שידוך חוקי M בין -

$$\{1,2,\dots m\},\{1,2,\dots n\}$$

כך ש-

- M שידוך (כל פוזיציה משודכת לכל היותר פעם אחת)
- אם $(i,j),(i',j')\in M$ אזי $i<i'\Rightarrow j<j'$.

כמה עולה שידוך?

1. פוזיציה שאינה משודכת תעלה $\delta>0$.
2. $(i,j)\in M$ תעלה $\alpha(i,j)$ (בדרך כלל $\alpha(x,x)=0$)

כאשר δ,α הן חלק מהקלט.

הבעיה -

נתונים X,Y,δ,α - המטרה היא למצא שידוך M בעל עלות מינימלית.

טענה 6.2 אם הזוג $(m,n)\notin M$ אזי x_m או y_n לא שודכו כלל ב-M

הוכחה: נניח בשלילה כי $(m,n)\notin M$ אבל $(m,j),(i,n)\in M$ כך ש-$i<m, j<n$ - זו היא סתירה לתנאי השני. ■

לכן, בפתרון אופטימלי M^* מתקיים לפחות אחד מהבאים -

1. $(m,n)\in M^*$
2. $m\in X$ לא שודך ב-M^*
3. $n\in Y$ לא שודך ב-M^*

נסמן $OPT(i,j)$ - עלות השידוך הקטנה ביותר של המחרוזות $x_1x_2\dots x_i,\ y_1y_2\dots y_j$. אזי -

$$OPT(i,j)=min\left\{\begin{array}{c}\alpha(x_i,y_j)+OPT(i-1,j-1),\\ \delta+OPT(i-1,j),\\ \delta+OPT(i,j-1)\end{array}\right\}$$

כאשר -

$$\begin{cases} OPT(i,0)=i\cdot\delta \\ OPT(0,j)=j\cdot\delta \end{cases}$$

בסך הכל ניתן לחשב עלות שידוך אופטימלי ב-$O(mn)$.

6.4 בעיית Subset Sum

נתונים n מספרים טבעיים $a_1, a_2 \ldots a_n$ ונתון מספר טבעי K.

השאלה - האם יש תת קבוצה $S \subseteq \{1, 2, \ldots n\}$ כך ש - $\sum\limits_{i \in S} a_i = K$.

נגדיר -

$P(i,k)$ - האם יש תת קבוצה של $\{a_1, a_2, \ldots a_i\}$ שסכומה בדיוק k?

נשים לב שמתקיים -

$$P(i,k) = P(i-1,k) \vee P(i-1, k-a_i)$$

כאשר -

$$P(i,k) = \begin{cases} False & k < 0 \\ False & i = 0, k \neq 0 \\ True & k = 0 \end{cases}$$

הסיבוכיות - $O(n \cdot k)$ כאשר הקלט הוא בגודל $O(n + log(k))$ - כלומר הסיבוכיות גרועה למדי.

6.5 בעיית Knapsack

נתונים n פריטים. לכל פריט i מוגדרים -

- s_i גודל הפריט ה-i.
- p_i הרווח של הפריט ה-i.

B - גודל התרמיל (כל הגדלים טבעיים).

נרצה למצא סט של פריטים $S \subseteq \{1, 2, \ldots n\}$ כך ש-$\sum\limits_{i \in S} p_i$ מקסימלי ו-$size(S) = \sum\limits_{i \in S} s_i \leq B$.

פתרון אחד לבעיה היא למיין את הפריטים לפי המשקל הסגולי (רווח ליחידת גודל) ולפעול באלגוריתם חמדני לפי המשקל הסגולי - פתרון כזה יעבוד בזמן פולינומיאלי בקלט, ויתן פתרון טוב ברוב המקרים - אבל לא יבטיח את אופטימליות הפתרון.

נגדיר כעת $X_{i,j}$ - תת הקבוצה הקטנה ביותר (מבחינת $size$) של הפריטים מתוך $\{1, 2, \ldots i\}$ שהרווח שלה הוא בדיוק j. אם לא קיימת כזו נגדיר $X_{i,j} = Null$. אזי -

$$X_{i+1,j} = \begin{cases} X_{i,j} & size\left(X_{i,j-p_{i+1}} \cup \{i+1\}\right) > B \quad or \quad X_{i,j-p_{i+1}} = Null \\ \min_{size}\left\{X_{i,j}, X_{i,j-p_{i+1}} \cup \{i+1\}\right\} & Otherwise \end{cases}$$

הערה 6.3 בהינתן שתי תתי קבוצות X ו-Y של$\{1, 2, \ldots i\}$ בעלות רווח j ניקח את הקבוצה עם ה-$size$ הקטן יותר.

7 זרימה[12]

דוגמא

רשת תקשורת שהיא גרף מכוון (קשת בין צומת שולח וצומת מקבל) לכל קשת מוגדר קצב מקסימלי שניתן להעביר עליה, נשאלת השאלה -

- מה הקצב המקסימלי שניתן לשלוח?
- איך שולחים את הידיעות ברשת בשביל לקבל קצב מקסימלי?

הגדרה 7.1 רשת זרימה היא (G,s,t,c) -

- G גרף מכוון
- $s,t\in V$ כאשר s בתפקיד מקור ו-t בתפקיד בור
- $c:E\to\mathbb{R}^+$ (פונקציית זרימה - כמה ניתן להזרים על קשת e - $c(e)$)

הגדרה 7.2 פונקציה $f:E\to\mathbb{R}$ תקרא פונקציית זרימה אם מתקיים -

- $0\le f(e)\le c(e)$ לכל $e\in E$ (אילוצי קיבול)
  - לכל צומת $v\neq s,t$ מתקיים -[13]

$$\sum_{e\in\delta^+(v)} f(e)=\sum_{e\in\delta^-(v)} f(e)$$

(אילוצי שימור)

הגדרה 7.3 ערך של פונקציית זרימה f יסומן ע"י $|f|$ ויוגדר להיות -

$$|f|\equiv\sum_{e\in\delta^+(s)} f(e)-\sum_{e\in\delta^-(s)} f(e)$$

("נטו" זרימה שיוצאת מ-s).

הבעיה

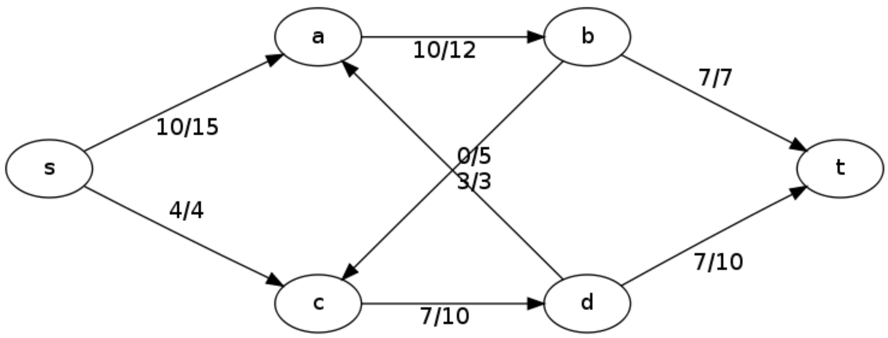
בהנתן רשת זרימה (G,s,t,c) רוצים למצא פונקציית זרימה f המביאה למקסימום את $|f|$.

הגדרה 7.4 חתך $\left(S,\bar{S}\right)$ (כאשר $S\subseteq V$) ייקרא חתך $s-t$ אם מתקיים -

$$s\in S,\ t\in\bar{S}$$

[12] הרצאה 12 - 28/12/09

[13] כאשר $\delta^+(v)$ היא קבוצת הקשתות היוצאות מהצומת v, ו-$\delta^-(v)$ קבוצת הקשתות הנכנסות לצומת v



איור 9: דוגמא לרשת זרימה חוקית עם $|f| = 14$

הערה 7.5 מעתה כל החתכים שנדון בהם יהיו חתכי $s-t$.

טענה 7.6 פונקציית זרימה ברשת (G,s,t,c) לכל חתך $\left(S,\bar{S}\right)$ מתקיים -

$$|f| = \sum_{\substack{e=(u,v)\in E\\ u\in s\\ v\in\bar{S}}} f(e) - \sum_{\substack{e=(u,v)\in E\\ u\in\bar{s}\\ v\in S}} f(e)$$

("נטו" זרימה שיוצאת מהחתך S היא בדיוק ערך פונקציית הזרימה f).

הוכחה: נסכים על אילוצי השימור לכל הצמתים ב-S -

נסתכל על אילוצי השימור ל-$s \neq v \in S$ כלשהו -

$$0 = \sum_{e\in\delta^{+}(v)} f(e) - \sum_{e\in\delta^{-}(v)} f(e)$$

לפי הגדרת $|f|$ נקבל -

$$|f| = \sum_{v\in S}\left(\sum_{e\in\delta^{+}(v)} f(e) - \sum_{e\in\delta^{-}(v)} f(e)\right)$$

קשת $e:(u\to v)$ כך ש-$u,v\in S$ תתבטל באגף ימין, כלומר בסך הכל נשאר עם -

$$|f| = \sum_{\substack{e=(u,v)\in E\\ u\in s\\ v\in\bar{S}}} f(e) - \sum_{\substack{e=(u,v)\in E\\ u\in\bar{s}\\ v\in S}} f(e)$$

■

הגדרה 7.7 קיבול של חתך $s-t$ $\left(S,\bar{S}\right)$ יסומן ע"י $c(S)$ -

$$c(S) \equiv \sum_{\substack{e=(u,v)\in E\\ u\in s\\ v\in\bar{S}}} c(e)$$

טענה 7.8 לכל חתך $\left(S,\bar{S}\right)$ מתקיים $|f|\leq c\left(S\right)$

הוכחה: לכל קשת $\left(u\rightarrow v\right)$ כך ש-$u\in S,\ v\notin S$ ו-$f\left(e\right)\leq c\left(e\right)$ (סכימה על אלו תיתן $c\left(S\right)$). לכל קשת $e:\left(u\rightarrow v\right)$ כך ש-$u\notin S$ ו-$v\in S$ מתקיים $f\left(e\right)\geq0$, לפי טענה 7.6 נקבל ש-$|f|\leq c\left(S\right)$. ■

הגדרה 7.9 עבור רשת זרימה $\left(G,s,t,c\right)$ ופונקציית זרימה f, נגדיר <u>רשת שיורית</u> באופן הבא -

$$\left(G_{f},s,t,c_{f}\right)$$

- $G_{f}=\left(V,E_{f}\right)$ כך ש-$E_{f}=E\cup\left\{ e^{R}:\left(v\rightarrow u\right)|e:\left(u\rightarrow v\right)\in E\right\}$
- לכל קשת $e\in E_{f}$ -

$$\begin{cases}
c_{f}\left(e\right)=c\left(e\right)-f\left(e\right)\\
c_{f}\left(e^{R}\right)=f\left(e\right)
\end{cases}$$

הגדרה 7.10 מסלול ברשת השיורית ייקרא <u>מסלול שיפור</u> אם הוא מתחיל ב-s ומסתיים ב-t ומשתמש בקשתות שה-קיבול השיורי שלהן חיובי ממש.

הגדרה 7.11 f פונקציית זרימה ברשת $\left(G,s,t,c\right)$ ו-f' פונקציית זרימה ברשת השיורית $\left(G_{f},s,t,c_{f}\right)$. פונקציית הזרימה $f+f'$ תוגדר באופן הבא -

$$\left(f+f'\right)\left(e\right)\equiv f\left(e\right)+f'\left(e\right)-f'\left(e^{R}\right)$$

הערה 7.12 $f+f'$ מוגדרת לרשת הזרימה $\left(G,s,t,c\right)$.

טענה 7.13 f פונקציית זרימה ב-$\left(G,s,t,c\right)$ ו-f' פונקציית זרימה ברשת $\left(G_{f},s,t,c_{f}\right)$. אז הפונקציה $f+f'$ מקיימת

- $f+f'$ היא פונקציית זרימה חוקית ב-$\left(G,s,t,c\right)$.
- $|f+f'|=|f|+|f'|$.

הוכחה: [14]

- נראה כי מתקיימים התנאים של זרימה חוקית -
  – אילוצי קיבול -

$$\left(f+f'\right)\left(e\right)=f\left(e\right)+f'\left(e\right)-f'\left(e^{R}\right)$$

[14] הרצאה 12 4/1/2010

כל אחת מהפונקציות f, f' הן פונקציות זרימה חוקיות, לכן $f(e), f'(e)$ מספרים חיוביים. כמו כן מהגדרת רשת זרימה שיורית מתקיים -

$$\begin{aligned} f'\left(e^R\right) &\leq f(e) \\ &\Downarrow \\ f(e) - f'\left(e^R\right) &\geq 0 \end{aligned}$$

ולכן -

$$(f+f')(e) = f(e) + f'(e) - f'\left(e^R\right) \geq 0$$

כמו כן מתקיים -

$$f'(e) \leq c(e) - f(e)$$

ולכן -

$$(f+f')(e) = f(e) + f'(e) - f'\left(e^R\right) \leq c(e)$$

– נראה אילוצי שימור, נבחר $v \neq s, e$ -

$$\begin{aligned} \text{total outgoing - } \sum_{e\in\delta^+(v)} (f+f')(e) &= \overbrace{\sum_{e\in\delta^+(v)} f(e)}^{(1)} + \overbrace{\sum_{e\in\delta^+(v)} f'(e)}^{(3)} - \overbrace{\sum_{e\in\delta^+(v)} f'\left(e^R\right)}^{(5)} \\ \text{total incoming-} \sum_{e\in\delta^-(v)} (f+f')(e) &= \underbrace{\sum_{e\in\delta^-(v)} f(e)}_{(2)} + \underbrace{\sum_{e\in\delta^-(v)} f'(e)}_{(4)} - \underbrace{\sum_{e\in\delta^-(v)} f'\left(e^R\right)}_{(6)} \end{aligned}$$

כיוון ש-f רשת זרימה מתקיים - $(1) = (2)$.

כיוון ש-f' רשת זרימה מתקיים שסך כל הזרימה שהיא מוציאה מצומת v ($(3)+(6)$) שווה לסך כל הזרימה שנכנסת לצומת ($(4)+(5)$) ובסה"כ -

$$\begin{aligned} (1)+(3)+(6) &= (2)+(4)+(5) \\ (1)+(3)-(5) &= (2)+(4)-(6) \end{aligned}$$

כלומר - שתי השורות שוות זו לזו.

- לפי הגדרה -

$$\begin{aligned} |f+f'| &= \sum_{e\in\delta^+(s)} (f+f')(e) - \sum_{e\in\delta^-(s)} (f+f')(e) = \\ &= \sum_{e\in\delta^+(s)} \left(f(e) + f'(e) - f'\left(e^R\right)\right) - \sum_{e\in\delta^-(s)} \left(f(e) + f'(e) - f'\left(e^R\right)\right) = \\ &= \underbrace{\sum_{e\in\delta^+(s)} f(e) - \sum_{e\in\delta^-(s)} f'(e)}_{|f|} + \underbrace{\sum_{e\in\delta^+(s)} \left(f'(e) - f'\left(e^R\right)\right) - \sum_{e\in\delta^-(s)} \left(f'(e) - f'\left(e^R\right)\right)}_{|f'|} = \\ &= |f| + |f'| \end{aligned}$$

המעבר האחרון - בדיוק לפי אותם שיקולים של סוף הסעיף הקודם - וזה אכן מה שרצינו להוכית.

■

משפט 7.14 משפט ה-Min cut Max flow

f פונקציית זרימה ברשת (G,s,t,c), 3 התנאים הבאים שקולים -

- f זרימת מקסימום ב-(G,s,t,c).
- ב-G_f אין משלולי שיפור.
- קיים חתך $\left(S,\bar{S}\right)$ כך ש-$|f|=c\left(S\right)$.

הוכחה:

- $(1)\Rightarrow(2)$ -

ניח בשלילה שיש מסלול שיפור p הרשת השיורית G_f מ-s ל-t. נגדיר פונקציית זרימה ברשת השיורית f' שמזרימה על קשתות p את - $\min\limits_{e\in p}\left\{c_f\left(e\right)\right\}$. (ברור כי f' פונקציית זרימה חוקית ברשת השיורית).

לפי טענה 7.13 קיבלנו פונקציית זרימה חדשה ברשת (G,s,t,c)שערכה -

$$|f+f'|=|f|+|f'|=|f|+\min_{e\in p}\left\{c_f\left(e\right)\right\}>|f|$$

בסתירה למקסימליות של f.

- $(2)\Rightarrow(3)$ -

נגדיר חתך $\left(S,\bar{S}\right)$ באופן הבא - ב-S קיימים כל הצמתים u מ-V כך שיש מסלול מ-s ל-u שעליו כל הקשתות עם קיבול שיורי חיובי ממש.

נשים לב כי $t\notin s$ (לפי הנתון כי אין כ-G_f מסלול שיפור), כמוכן $s\in S$ (לפי המסלול הריק מ-s לעצמו) כלומר $\left(S,\bar{S}\right)$הוא חתך $s-t$.

נתובנן על קשת חתך $e:u\to v$ כלשהי בחתך $\left(S,\bar{S}\right)$ הנ"ל.

- אם $v\in\bar{S},\ \ u\in S,$ בהכרח מתקיים $f\left(e\right)=c\left(e\right)$ (כי אחרת ב-G_f היתה קשת $u\to v$ עם קיבול שיורי חיובי ממש ואז $v\in S$).
- אחרת - $v\in S,\ \ u\in\bar{S},$ בהכרח מתקיים $f\left(e\right)=0$. אם $f\left(e\right)>0$ אזי ברשת השיורית G_f הייתה $e^R:v\to u$ עם קיבול שיורי חיובי ממש ואז יש מסלול ברשת השיורית G_f מ-s ל-u המשתמש בקשתות עם קיבול שיורי חיובי ממש - בסתירה לכך ש-$u\notin S$.

לפי טענה 7.6 מתקיים -

$$|f|=\sum_{\substack{e=(u,v)\in E\\ u\in s\\ v\in\bar{S}}}f\left(e\right)-\sum_{\substack{e=(u,v)\in E\\ u\in\bar{s}\\ v\in S}}f\left(e\right)=\sum_{\substack{e=(u,v)\in E\\ u\in s\\ v\in\bar{S}}}c\left(e\right)-0=c\left(S\right)$$

- $(3)\Rightarrow(1)$

ניח בשלילה ש-f אינה זרימת מקסימום. לכן קיימת f^* ברשת (G,s,t,c) כך ש-

$$|f^*|>|f|=c\left(S\right)$$

אבל לפי טענה 7.8 מתקיים $|f^*|\le c\left(S\right)$ וזו סתירה.

■

7.1 האלגוריתם של Ford-Fulkerson

1. מאתחלים את $f(e)=0$ לכל $e\in E$.
2. כל עוד ב-G_f יש מסלול שיפור p -
   (א) תהא f' פונקציית הזרימה ב-G_f המזרימה על p את $\min\limits_{e\in p}\{c_f(e)\}$ (ואפס על קשתות אחרות).
   (ב) $f\leftarrow f+f'$
3. החזר את f

נכונות

נכונות האלגורימם נובעת ממשפט ה-Min cut Max flow.

הערה 7.15 אם מאפשרים קיבולים לא רציונליים האלגוריתם לא בהכרח יעצור, לא ניתן לייצג קיבלוים כאלו בביטים ולכן נניח שהקיבלוים שלמים (אם רציונליים - נכפול את כולם במספר כזה כך שיהפכו לשלמים).

הערה 7.16 אם f^* היא זרימת מקסימום כלשהי אז האלגוריתם עלול לעשות $|f^*|$ איטרציות. בכל איטרציה יש לבדוק האם קיים מסלול שיפור ולכן סיבוכיות האלגוריתם $O(e\cdot|f^*|)$.

שאלה

האם ניתן לבחור מסלולים בצורה "חכמה" בשביל לשפר את זמן הריצה?

הצעה - לבחור מסלול שיפור שמגדיל את ערך הזרימה בערך הכי גדול.

הגדרה 7.17 f פונקצית זרימה ברשת (G,s,t,c) פירוק של f ל-k מסלולי זרימה הוא k פונקציות זרימה - $f_{p_1},f_{p_2},\dots f_{p_k}$ כך ש-

- p_i הוא מסלול מ-s ל-t ברשת (G,s,t,c)
- f_{p_i} מזרימה זרימה חיובית רק על קשתות המסלול p_i
- f היא סכום הזרימות $f(e)=\sum_{i=1}^{k}f_{p_i}(e)$.

טענה 7.18 אם f פונקציית זרימה ברשת (G,s,t,c) אז ל-f קיים פירוק למסלולי זרימה המכיל לכל היותר $|E|$ מסלולי זרימה.

טענה 7.19 f פונקציית זרימה ברשת (G,s,t,c), f^* זרימת מקסימום ברשת (G,s,t,c). אז ערך זרימת המקסימום ברשת השיורית (G_f,s,t,c_f) הוא - $|f^*|-|f|$.

	זרימה f	זרימת מקסימום ב-G_f	השיפור

צעד 1	0	$\lvert f^*\rvert$	$\lvert f_1\rvert \ge \frac{\lvert f^*\rvert}{\lvert E\rvert}$ (לפי שתי הטענות הנ"ל)
צעד 2	f_1	לכל היותר $\left(1-\frac{1}{\lvert E\rvert}\right)\lvert f^*\rvert$	$\lvert f_2\rvert \ge \frac{\left(1-\frac{1}{\lvert E\rvert}\right)\lvert f^*\rvert}{\lvert E\rvert}$
צעד 3	f_1+f_2	לכל היותר $\left(1-\frac{1}{\lvert E\rvert}\right)^2\lvert f^*\rvert$	...
צעד k	...	לכל היותר $\left(1-\frac{1}{\lvert E\rvert}\right)^{k-1}\lvert f^*\rvert$	

השאלה - מה הוא ערך k עבורו $|f^*|\left(1-\frac{1}{|E|}\right)^{k-1}$ קטן או שווה ל-1?
אחרי $|E|+1$ איטרציות נשאר עם ערך זרימת מקסימום ברשת השיורית -

$$\left(1-\frac{1}{|E|}\right)^{|E|}|f^*| \le \frac{1}{e}|f^*|$$

כלומר, הסיבוכיות - $O\left(|E|\,log\left(|f^*|\right)\right)$ איטרציות.

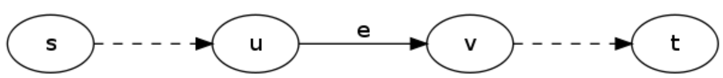
הגדרה 7.20 הגרעין[15] של רשת זרימה (G,s,t,c) הוא -

$$S\left(G\right)=\left\{e\in E|\begin{array}{c}e\text{ on a shortest path}\\ \text{from }s\text{ to }t\text{ in }G\end{array}\right\}$$

הערה 7.21 אנו נתעניין בסופו של דבר בגרעין של הרשת השיורית.

טענה 7.22 תהא (G,s,t,c) רשת זרימה. תהא $e=(u\to v)\in E$ הנמצאת בגרעין $S\left(E\right)$. אם נוסיף את הקשת $e^R=(v\to u)$ הגרעין לא ישתנה.

הוכחה: $\delta_G\left(x,y\right)$ סימון לאורך המסלול הקצר ביותר מ-x ל-y ב-G. נראה ש-$\delta_G\left(s,t\right)$ לא משתנה אחרי הוספת הקשת e^R. יהא P מסלול קצר ביותר מ-s ל-t המשתמש בקשת e (קיום P מובטח כי $e\in S\left(G\right)$)



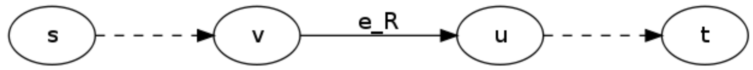
איור 10: המסלול P באופן סכמטי

$$\delta_G\left(s,t\right)=\delta_G\left(s,v\right)+\delta_G\left(u,t\right)-1$$

(בגלל שתת מסלול של מסלול קצר ביותר גם הוא קצר ביותר) נניח בשלילה כי הוספת הקשת e^R ל-G הקטינה את $\delta_G\left(s,t\right)$ - אז קיים מסלול קצר ביותר חדש P' המשתמש ב-e^R -
אורך P' -

$$\delta_G\left(s,v\right)+\delta_G\left(u,t\right)+1$$

[15] הרצאה 12 - 11.1.10



איור 11: המסלול P' באופן סכמטי, יש לשים לב כי u,v החליפו את מקומן

קיבלנו כי האורך של P' גדול ממש מ-$\delta_G(s,t)$ - סתירה.

כלומר הוספת קשת e^R אינה משנה את $\delta_G(s,t)$. כל מסלול קצר ביותר מ-s ל-t לפני הוספת הקשת e^R נשאר מסלול קצר ביותר גם אחרי הוספת הקשת.

כל קשת שהייתה בגרעין לפני הוספת e^R נשארה בגרעין אחרי הוספת e^R.

נשים לב כי לא ייתכן שהוספנו קשתות חדשות לגרעין כי אז הוספת e^R גרמה להווצרות מסלול קצר ביותר חדש מ-s ל-t (וראינו שזה בלתי אפשרי). ■

מסקנה 7.23 אם נוסיף k קשתות הפוכות לקשתות גרעין הגרעין לא ישתנה (ניתן להוכיח באינדוקציה).

טענה 7.24 יהיו G' ו-G'' שתי רשתות שיוריות עוקבות במהלך ריצת Edmonds Karp. אז לפחות אחד מהשניים הבאים קורה -

1. $\delta_{G'}(s,t) < \delta_{G''}(s,t)$
2. $|S(G'')| < |S(G')|$

הוכחה: מה קורה כאשר G'' נוצרת מ-G'?

1. נוספות קשתות הפוכות לקשתות ב-$S(G')$
2. לפחות קשת גרעין אחת של $S(G')$ תעלם (הקיבול השיורי שלה יהפוך לאפס)

כאן משתמשים בעובדה שהאלגוריתם בוחר מסלול שיפור קצר ביותר ב-G' (כלומר המסלול מכיל רק קשתות גרעין $S(G')$). לצורך הניתוח "נבצע" סדרתית קודם את (1) (הוספת קשתות הפוכות לגרעין $S(G')$) ואחר כך את (2).

נראה כי $\delta_{G'}(s,t) \leq \delta_{G''}(s,t)$ -

לפי המסקנה לעיל אחרי הוספת הקשת ההפכית ל-$S(G')$ ב-(1) אורך המסלול הקצר ביותר מ-s ל-t נשמר.

בשלב (2) רק מוציאים קשתות מהגרעין לכן פעולה זו יכולה רק להגדיל את אורך המסלול הקצר ביותר מ-s ל-t.

$\Leftarrow \delta_{G'}(s,t) \leq \delta_{G''}(s,t)$.

אם $\delta_{G'}(s,t) < \delta_{G''}(s,t)$ סיימנו. אחרת - נניח ש-$\delta_{G'}(s,t) = \delta_{G''}(s,t)$

$\Leftarrow$ אחרי שלב (1) לפי המסקנה $S(G') = S(G'')$. אחרי שלב (2) (תחת ההנחה ש-$\delta_{G'}(s,t) = \delta_{G''}(s,t)$) נקבל ש-$S(G'') \subsetneq S(G')$ לכן - $|S(G'')| < |S(G')|$. ■

לפי משפט ה-Min cut Max flow כאשר אין מסלול מ-s ל-t ברשת השיורית (לחילופין - הגרעין של הרשת השיורית ריק) יש לנו זרימת מקסימום.

$\Leftarrow$ כמות האיטרציות $O(|V||E|)$. סה"כ - $O\left(|V||E|^2\right)$ (כל איטרציה מריצה BFS ברשת השיורית).

7.1.1 האלגוריתם של דייניץ

מבצע את המשימה ב-$O\left(|V|^2|E|\right)$. פועל בצורה דומה לאלגוריתם של אדמונדס וקארפ אבל מבצע בכל שלב.

7.2 שידוך גדול ביותר בגרף דו צדדי

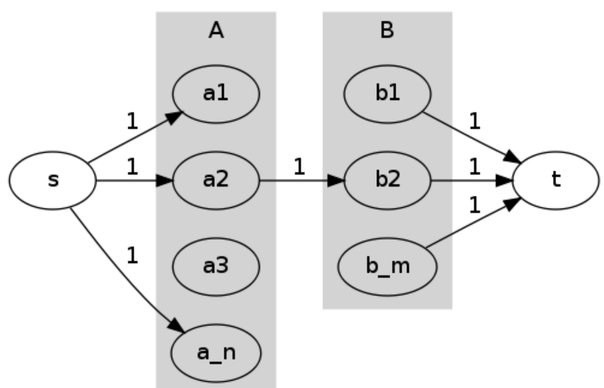
הגדרה 7.25 בהנתן גרף לא מכוון $G=(V,E)$ שידוך הוא תת קבוצה של קשתות $X\subseteq E$ כל שלכל שתי קשתות ב-X אין צומת קצה משותף.

הבעיה

$G=(A,B,E)$ גרף דו צדדי. צריך למצא שידוך גדול ביותר ב-G.

פתרון

נבנה את רשת הזרימה באופן הבא -



איור 12: בניית רשת הזרימה באופן סכמטי

פורמלית -

$$(G',s,t,c')$$

$$\begin{cases} G'= & (V',E') \\ V'= & A\cup B\cup\{s,t\} \\ E'= & \{s\to w|w\in A\}\cup\{v\to t|v\in B\}\cup\{u\to v|u\in A,\ v\in B,\ u\to v\in E\} \end{cases}$$

$$c'(e')=1\ \ \forall e'\in E$$

טענה 7.26 ברשת זרימה (G,s,t,c) בעלת קיבולים שלמים $c(e)\in\mathbb{N}$ (לכל $e\in E$) קיימת זרימת מקסימום שלמה וניתן למצא אותה.

הערה 7.27 אומרית ש-f זרימה שלמה אם $f(e) \in \mathbb{N}$ לכל $e \in E$.

הוכחה: צריך להראות שבתחילת כל איטרציה של FF מתקיים -

- f שלמה
- כל הקיבולים השיוריים שלמים

ההוכחה באינדוקציה של האיטרציות של FF .

■

מסקנה 7.28 ברשתשבנינו על כל קשת יזרום בזרימת מקסימום 0 או 1.

7.2.1 האלגוריתם

1. בנה את רשת הזרימה (G', s, t, c') כנ"ל.
2. מצא זרימת מקסימום $f*$ ברשת (G', s, t, c')
3. הוסף ל-X כל קשת מ-G': $(u \to v)$ כך ש-

$$u \in A,\ v \in B\ f(u \to v) = 1$$

הוכחת נכונות

טענה 7.29 X הוא שידוך חוקי.

הוכחה: נניח בשלילה שב-X יש צומת $u \in A$ שנוגעת בשתי קשתות.
$\Leftarrow$ כמות הזרימה שיוצאת מ-u היא לפחות 2, מאילוצי שימור של f^* ל-u נכנסות לפחות 2 יחידות זרימה. אבל מבניית G' יש קשת יחידה שנכנסת ל-u $(s \to u)$ וקיבולה בדיוק 1. סתירה.
המקרה שיש צומת $v \in B$ שנוגעות בו שתי קשתות - סימטרי. ■

טענה 7.30 $|X| = |f^*|$

הוכחה: לפי המסקנה $\Leftarrow$
לכל קשת $(u \to v) \in E'$ מתקיים $f(u \to v) = 1$ או $f(u \to v) = 0$. מהבנייה - $|X|$ הוא מספר הקשתות שעליהן הזרם הוא 1.
נסתכל על החתך הבא -

$$\begin{cases} S = & \{s\} \cup A \\ \bar{S} = & \{t\} \cup B \end{cases}$$

הזרימה של f על פני החתך $(S, \bar{S})$ היא $|X|$ לכן - $|X| = |f^*|$. ■

טענה 7.31 X הוא שידוך מקסימום. הוכחה: נניח בשלילה שקיים ב-G שידוך X' כך ש-

$$|X'| > |X|$$

נבנה פונקציית זרימה f' באופן הבא -

עוברים על קשתות $e \in X'$ אחת אחת. לכל קשת $e : u \to v$ הוסף ל-f' יחידת זרימה על $(u \in A, v \in B)$. הקשתות

$$(s \to u)$$

$$(u \to v)$$

$$(v \to t)$$

נראה ש-f' פונקציית זרימה חוקית ברשת (G', s, t, c') - ■

- אילוצי שימור - מאופן בניית f' בכל איטרציה מתקיים שימור לכל הצמתים שאינם s ו-t.
- אילוצי קיבול - כיוון ש-X' הוא שידוך חוקי כל צומת $u \in A$ נוגעת לכל היותר קשת אחת ב-X'....

קיבלנו רשת זרימה f' שמזרימה יותר מ-$f*$ - זו סתירה...

8 כפל מהיר של פולינומים[16]

$A(x)$ הוא פולינום ממעלה n אם -

$$A(x) = \sum_{j=0}^{n} a_j x^j$$

ומתקיים - $a_n \neq 0$.

כיצד מייצגים פולינום?

- ייצוג 1 - באמצעות וקטור המקדמים - $\vec{a} = (a_0, a_1, \ldots a_n)$
  - להעריך את הפולינום A בנקודה x_0 -
    באופן נאיבי צריך $O(n^2)$ פעולות, אבל אפשר לחשב לפי סכמת הורנר -

    $$A(x_0) = a_0 + x_0 (a_1 + x_0 (a_2 + \ldots))$$

    בסה"כ - $O(n)$ פעולות אריתמטיות.
  - לחבר שני פולינומים - נחבר את המקדמים, $O(n)$ פעולות.

[16] הרצאה 14 - 18/1/10 - לא מלאה

– כפל פולינומים -

$$C\left(x\right)=A\left(x\right)\cdot B\left(x\right)$$

במקרה זה -

$$c_{j}=\sum_{k=0}^{j}a_{k}b_{j-k}$$

לכן חישוב המכפלה יתבצע בסיבוכיות $O\left(n^{2}\right)$.

- ייצוג 2 - פולינום ממעלה n ייוצג ע"י ערכיו ב-$n+1$ נקודות שונות.

$$\left\{ \left(x_{0},y_{0}\right),\left(x_{1},y_{1}\right)\ldots,\left(x_{n+1},y_{n+1}\right)\right\} \ A\left(x_{i}\right)=y_{i}\ \forall i=0,1,\ldots n$$

איך עוברים מייצוג 1 לייצוג 2? בוחרים $n+1$ ערכים שונים, עבור כל ערך מחשבים את ההצבה שלו בפולינום. סה"כ - $O\left(n^{2}\right)$ פעולות.

אינטרפולציה

לכל ייצוג $\left\{ \left(x_{0},y_{0}\right),\left(x_{1},y_{1}\right)\ldots,\left(x_{n+1},y_{n+1}\right)\right\}$ כך ש-$x_{i}\neq x_{j}$ לכל $i\neq j$ קיים פולינום יחיד $A\left(x\right)$ ממעלה לכל היותר n כך ש-$A\left(x_{i}\right)=y_{i}$ לכל $i=0,1,\ldots n$. **הוכחה:** אפשר לפתוח מחברת אנליזה נומרית... ∎

מסקנה 8.1 ניתן לעבור מייצוג 2 לייצוג 1 ע"י פתרון מערכת משוואות לינארית, נאיבית בסיבוכיות - $O\left(n^{3}\right)$ וע"י שימוש בשיטת לגראנז - $O\left(n^{2}\right)$.

חיבור שני פולינומים ממעלה לכל היותר n שמיוצגים בשיטה 2 -

- אם $A\left(x\right)$ ו-$B\left(x\right)$ מיוצגים באותם $n+1$ נקודות (ערכי x זהים) ניתן לחשב פשוט -

$$C\left(x_{j}\right)=A\left(x_{j}\right)+B\left(x_{j}\right)$$

וניתן לחשב ב-$O\left(n\right)$.

מכפלה של שני פולינומים ממעלה n לכל היותר -

- אם נניח כי A ו-B מיוצגים על ידי אותן $2n+1$ נקודות (ערכי x זהים) אפשר פשוט לכפול נקודתית -

$$C\left(x_{j}\right)=A\left(x_{j}\right)\cdot B\left(x_{j}\right)$$

לכן במקרה זה ניתן לכפול ב-$O\left(n\right)$.

8.1 הרעיון

נתונים $A\left(x\right)$ ו-$B\left(x\right)$ בייצוג הראשון, נרצה לעבור לייצוג השני, לכפול בייצוג זה ולחזור לייצוג הראשון. ע"י בחירת שערוך A,B בשורשי היחידה המרוכבים (מסדר $2n+1$) ניתן לממש את המעבר בין הייצוג בזמן -

$$O\left(n\cdot log\left(n\right)\right)$$