תזכורת

נתחיל בתזכורת למה שלמדנו לפני שכל האוכל הנורמלי נעלם מהמדינה ואנשים הוכרחו לאכול חתיכות דיקט בגודל של מסך מחשב.

האלגוריתם הגנרי למציאת מק"ב ממקור יחיד בגרף מכוון ממושקל

<u>מתחזקים</u> לכל $v \in V$:

- $d[v]$ – אורך מסלול כלשהו מ- s (או ∞)
- $\pi[v]$ – ההורה של v בעץ מק"ב מ- s (יכול להיות Null)

<u>האלגוריתם</u>:

- $\text{Relax}(u,v)$:
  - אם $d[v] > d[u] + w(u,v)$ אזי:
    - $d[v] \leftarrow d[u] + w(u,v)$
    - $\pi[v] \leftarrow u$
- <u>אתחול</u>:
  - $d[s] \leftarrow 0$
  - $d[v] \leftarrow \infty$ לכל $v \neq s$
  - $\pi[v] \leftarrow \text{Null} \quad \forall v \in V$
- <u>צעד</u>: נבחר קשת (u,v) ונבצע $\text{Relax}(u,v)$
- <u>תנאי סיום</u>: לכל קשת (u,v) מתקיים $d[v] \leq d[u] + w(u,v)$ ולכן אי-אפשר לשנות עוד ערכים באמצעות Relax.

<u>הוכחנו</u>: <u>משפט נכונות האלגוריתם הגנרי</u>:

- תנאי הסיום מתקיים $\Leftrightarrow d[v] = \delta(s,v)$
- אם התנאי הנ"ל מתקיים אזי הגרף $\left(V, \left\{(\pi[v], v) \middle| v \neq s\right\}\right)$ מהווה עץ מק"ב מ- s לכל הקדקודים.

האלגוריתם של Bellman-Ford (עבור משקולות כלליים)

- <u>אתחול</u> כמו באלגוריתם הגנרי
- <u>לולאת צעדי Relax</u>:
  - $|V|-1$ פעמים נעבור על כל הקשתות ונבצע $\text{Relax}(u,v)$ לכל קשת (u,v)
- <u>בדיקת מעגלים שליליים באמצעות תנאי הסיום</u>:
  - נבדוק את תנאי הסיום באלגוריתם הגנרי:
    - אם הוא מתקיים, נחזיר את הנתונים $\pi[v], d[v]$ לכל v
    - אם לא, נודיע "קיים מעגל שלילי בגרף".

<u>הוכחנו</u>: <u>טענה</u>: אם קיים מעגל שלילי, האלגוריתם מודיע זאת.

<u>ניסחנו</u>: <u>משפט</u>: אם אין מעגל שלילי, האלגוריתם של בלמן ופורד מחזיר מרחקים ק"ב ועץ מק"ב.

הוכחנו את המשפט באמצעות:

[המשך החומר] טענת עזר

אם אין מעגלים שליליים, אזי בסוף האיטרציה ה- k, לכל קדקוד $v \in V$ כך שקיים מק"ב מ- s ל- v בעל $\geq k$ קשתות, מתקיים כי $d[v] \leq \delta(s,v)$.

[ציור]

<u>הוכחה</u>:

באינדוקציה על k:

- <u>מקרה בסיס</u>: $k=0$
  מסלול עם 0 קשתות מ- s קיים רק עבור $v=s$.
  עבור s, מתקיים $\delta(s,s)=0$ (כי אין מעגלים שליליים), ולכן אחרי האתחול מתקיים $d[s] \leq \delta(s,s)$.

ריבוע ☐

- <u>צעד אינדוקטיבי</u>: נניח עבר כל $k'<k$ ונוכיח את הטענה עבור k.
  יהי $v \in V$ קדקוד כלשהו כך שקיים מק"ב p מ- s ל- v עם k קשתות לכל היותר.
  - <u>מקרה 1</u>:
    אם $|p|<k$ אז לפי הנחת האינדוקציה, בשלב קודם באלגוריתם כבר התקיים $d[v] \leq \delta(s,v)$, והאלגוריתם יכול רק להקטין את $d[v]$ ע"י Relax – לכן אי-השוויון ממשיך להתקיים.
  - <u>מקרה 2</u>:
    אם p מכיל בדיוק k $(1\leq)$ קשתות, תהי (u,v) הקשת האחרונה ב- p, כלומר $p=p'\circ(u,v)$ עבור מסלול p' מ- s ל- u בעל $k-1$ קשתות.
    [ציור]
    לפי טענה קודמת (תת-מסלול של מק"ב הוא בעצמו מק"ב), המסלול p' הוא מק"ב מ- s ל- u, ומכיל $k-1$ קשתות.
    לכן, לפי הנחת האינדוקציה, בסוף האיטרציה ה- $(k-1)$ כבר התקיים

$$\boxed{d[u] \leq \delta(s,u)} \quad \boxed{(*)}$$

שוב, כיוון ש- Relax רק מקטין ערכי $d[\cdot]$, אי-שוויון (*) התקיים בכל מהלך האיטרציה ה- k.
בפרט, אחרי ביצוע $\text{Relax}(u,v)$ באיטרציה ה- k, התקבל אי-השוויון:

$$d[v]\le d[u]+w(u,v)\overset{(*)}{\le}\delta(s,u)+w(u,v)=w(p')+w(u,v)=w(p)=\delta(s,v)$$

זה אי-השוויון שרצינו להוכיח, ומשום ש- Relax רק יכול להקטין את $[\cdot]$, אי-שוויון זה ממשיך להתקיים עד סוף האיטרציה ה- k.

□

זמן ריצה

מבצעים כ- $|V|$ איטרציות וכ- $|E|$ פעולות באיטרציה, לכן:

$$O(|V|\cdot|E|)$$

האלגוריתם של Physarum Polycephalum

בגרף לא מכוון עם משקל מקסימלי W מוצא מק"ב מ- s ל- t כד כדי $(1+\varepsilon)$ בזמן:

$$O\left(|E|\cdot W\cdot(\log|V|+\log W)\frac{1}{\varepsilon^2}\right)$$

[QR Code]

[לא ניכנס לזה, כי הוא ממומש ע"י ג'יפה צהובה.]

מסלולים קצרים ביותר בין כל הזוגות (All-pairs shortest paths)

[הפעם נרצה למצוא את כל המסלולים הקצרים ביותר בין **כל** שני קדקודים בגרף.]

- הבעיה: מציאת מק"ב בין כל זוג קדקודים $u,v\in V$.

<u>רעיון לפתרון – רדוקציה למה שאנחנו כבר יודעים לעשות:</u>
לכל קדקוד v נריץ אלגוריתם למציאת מק"ב מ- v.

- עבור משקלות $0\le$ ⇐ $|V|$ פעמים דייקסטרה

⇐ זמן ריצה = $O\left(|V||E|+|V|^2\log|V|\right)=O\left(|V|\cdot T(\text{Dijkstra})\right)$

- המקרה הכללי (משקלות $\ni\mathbb{R}$) ⇐ $|V|$ פעמים Bellman-Ford

⇐ זמן ריצה: $O\left(|V|^2|E|\right)$

נראה בהמשך כיצד ניתן בכל-זאת להפעיל את האלגוריתם של דייקסטרה, אבל כעת נציע אלגוריתם שונה לחלוטין, שלא מבוסס על האלגוריתם הגנרי כלל.

האלגוריתם של Floyd-Warshall

- הרעיון: תכנון דינאמי.
  נסמן את הקדקודים ב- $V=\{1,2,\ldots,|V|\}$.
- תת-בעיות: לכל $i,j\in V$ ולכל $k\in V$ נגדיר:

$$\delta_{ij}^{k}:=\begin{cases}\text{אורך מסלול קצר ביותר מ-} i \text{ ל-} j \text{ שקדקודי הביניים שלו שייכים ל-} \{1,\ldots,k\} \\ \text{אם אין מסלול כזה} - \infty\end{cases}$$

- הבעיה שרוצים לפתור: (לכל $i,j\in V$)

$$\delta(i,j)=\delta_{ij}^{|V|}$$

נתבונן במקרים האפשריים ע"מ לבנות את נוסחת המבנה:

- מקרה בסיס: $k=0$

$$\delta_{ij}^{0}=\begin{cases}0, & i=j\\ w(i,j) & (i,j)\in E\\ \infty & \text{otherwise}\end{cases}$$

[[הסבר: במקרה זה אנו בוחנים מה משקל מק"ב בין i ל- j שלא עובר בדרך בשום קדקוד בגרף. אם $i=j$ קיים המסלול המנוון (i) שמשקלו 0. אם i ו- j שכנים, קיים המסלול שמכיל רק את הקשת שביניהם (כך לא עוברים בקדקודים בגרף כלל פרט לקדקוד הראשון והאחרון, שאותם אנו לא מחשיבים).]]

- מקרה כללי: $k\geq 1$
  אם אנו מניחים שאין מעגלים שליליים אז אין סיבה לחזור על קדקוד פעמיים במסלול מ- i ל- j (זה לא יכול **להקטין** את המשקל הכולל).
  לכן בפרט במסלול קצר ביותר מ- i ל- j אנו עוברים לכל היותר פעם אחת בקדקוד k.
  לכן יש שתי אפשרויות:
  - לא עוברים ב- k כלל:
    [ציור]
    במקרה זה אורך המסלול הוא δ_{ij}^{k-1}.
  - עוברים ב- k פעם אחת בדיוק:
    [ציור]
    במקרה זה ניתן לפרק את המסלול לשני חלקים: עד ל- k והחל מ- k.
    לכן אורך המסלול הוא $\delta_{ik}^{k-1}+\delta_{kj}^{k-1}$.

לכן המסלול הקצר ביותר מ- i ל- j שמשתמש אך ורק בקדקודים $1,\ldots,k$ הוא הקצר מבין שני הנ"ל, כלומר:

$$\delta_{ij}^{k} = \min\left\{\delta_{ij}^{k-1}, \delta_{ik}^{k-1} + \delta_{kj}^{k-1}\right\}$$

אלגוריתם איטרטיבי

- אתחול:
  - לכל i: $d_{ii}^{0} \leftarrow 0$
  - לכל קשת (i,j): $d_{ij}^{0} \leftarrow w(i,j)$
  - לכל $i \neq j$ כך ש- $(i,j) \notin E$: $d_{ij}^{0} \leftarrow \infty$
- לולאה מרכזית:
  - עבור k מ-1 ועד $|V|$, נריץ לולאה:
    - לכל $i,j \in V$:
      - $d_{ij}^{k} \leftarrow \min\left\{\delta_{ij}^{k-1}, \delta_{ik}^{k-1} + \delta_{kj}^{k-1}\right\}$

בניית עצים

לכל $i \in V$ נגדיר גרף T_i שיהווה מק"ב מ- i:

$$T_i = \left(V, \left\{\left(\pi_i[j], j\right) \middle| j \neq i\right\}\right)$$

- <u>אתחול</u>:

  נחשב שדות ביניים:

$$\pi_i^{0}[j] \leftarrow \begin{cases} i & (i,j) \in E \\ \text{Null} & \text{otherwise} \end{cases}$$

- <u>לולאה</u>:

  לכל $k = 1,\ldots,|V|$:

$$\pi_i^{k}[j] \leftarrow \begin{cases} \pi_i^{k-1}[j] & d_{ij}^{k-1} \leq d_{ik}^{k-1} + d_{kj}^{k-1} \\ \pi_k^{k-1} & \text{otherwise} \end{cases}$$

[להסביר]

זמן ריצה

יש שלושה אינדקסים שרצים על $|V|$ אפשרויות, לכן הסיבוכיות הכוללת היא $O\left(|V|^3\right)$ [וזה הכי טוב שידוע כיום].

האלגוריתם של Johnson

[זה אלגוריתם שנועד בעיקר להעשרה; הוא בסילבוס אבל הוא לא עניין מהותי בחומר.]

נרצה למצוא דרך להריץ את האלגוריתם של דייקסטרה, כי הוא נחמד ויעיל ומגניב ומזכיר לי את המילה ההולנדית ל-"דולפינים" (Dolfijn).

לצערנו, תיתכנה משקלות שליליות בגרף, וזה מפריע.
אנו יודעים שלא ניתן באופן נאיבי להוסיף קבוע לכל המשקלות, כי אז מסלולים עם כמויות שונות של קשתות משתנים באופן לא סימטרי.

האלגוריתם של Johnson בא לפתור בעיה זו.
אופן הפעולה:

- נוסיף קדקוד חדש לגרף, q, ונחבר אותו לכל הקדקודים עם קשתות במשקל 0.
- נריץ B-F [[= Bellman-Ford]] על הגרף החדש ממקור q ונקבל מרחקים $d'[v]$ לכל $v \in V$ (או שנגלה מעגל שלילי).
- אם אין מעגל שלילי, הערכים d' מקיימים:

$$d'[v] \le w(u,v) + d'[u]$$

לכל קשת (u,v).
- נגדיר:

$$w'(u,v) := w(u,v) + d'[u] - d'[v] \ge 0$$

- שלב אחרון: נריץ Dijkstra מכל קדקוד עם המשקולות החדשים $w'(\cdot\,,\cdot)$.
- אבחנה:
אם $p = (v_1, v_2, \dots, v_k)$ מסלול ב-G אזי:

$$\begin{aligned} w'(p) &= \sum_{i=1}^{k-1} w'(v_i, v_{i+1}) = \\ &= \sum_{i=1}^{k-1} \left(w(v_i, v_{i+1}) + d'[v_i] - d'[v_{i+1}] \right) = \\ &= \underbrace{\sum_{i=1}^{k-1} w(v_i, v_{i+1})}_{w(p)} + d'[v_1] - d'[v_k] \end{aligned}$$

[כלומר קיבלנו שהמשקל החדש של **כל** מסלול מ-v_1 ל-v_k שווה למשקל הקודם ועוד הערך $d'[v_1] - d'[v_k]$. כיוון שהקדקוד הראשון והאחרון במסלולים מ-v_1 ל-v_k קבועים, למעשה משקל כל המסלולים גדל בקבוע, לכן מק"ב עם המשקל החדש הוא גם מק"ב עם המשקל הישן.]