אלגוריתמים 1 הרצאה 5: $shorts\ path - APSP$

4 בדצמבר 2025

גיא יער־און

0.1 מבוא לכפל מטריצות מהיר

קלט: שתי מטריצות $A, B \in \mathbb{F}^{n\times n}$. נסמנן: $A = \{a_{ij}\}, B = \{b_{ij}\}$
פלט: לחשב את $C = A \times B$, כלומר את $C = \{c_{ij}\}$

דרך נאיבית לפתרון הבעיה: $C_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj}$, כל תא יעלה לחישוב לפי הנוסחה $O(n)$ והמטריצה בגודל $n \times n$ כלומר ישנם n^2 תאים ומכאן עלות האלגוריתם $O(n^3)$.

אחת מהשאלות החשובות בעולם התאוריה של מדעי המחשב, היא מהו הזמן הכי מהיר שבו ניתן לחשב את המטריצה C.

הבחנה ראשונה: זמן מינימלי לפתרון הבעיה הינו $\Omega(n^2)$ כיוון שגודל הפלט הינו $O(n^2)$. וכן הוא לינארי בגודל הקלט שלו.

האלגוריתם של שטרסן: רץ בזמן $O(n^{log_2 7}) = O(n^{2.81})$ - האלגוריתם שביצענו בעצמנו בתרגיל (1) שאלה (1), מגדירים כפלים חדשים ומצליחים להגיע לנוסחת הנסיגה $T(n) = 7T(\frac{n}{2}) + O(n^2)$.

בשנת 1986 נפל דבר, $Coppersmith–Winograd$ו הצליחו להגיע לסיבוכיות זמן של $O(n^{2.376})$.
לאחר מכן הסיבוכיות ירדה ל$O(n^{2.37287})$

נסמן בω את האקספוננט של האלגוריתם הכי מהיר שקיים לפתרון הבעיה. בהכרח, $\omega \leq 2.37287$ כי כיום ידוע אלגוריתם שפותר בזמן זה, וכן נקבל סה"כ כי:

$$2 \leq \omega \leq 2.37287$$

מכאן, האלגוריתם הכי מהיר לכפל מטריצות ריבועיות מגודל $n \times n$ עולה $O(n^\omega)$ זמן.

אנחנו לא נלמד על כפל מטריצות מהיר בקורס. עם זאת, ישנם הרבה אלגוריתמים שמשתמשים בכפל מטריצות מהיר בתור פרוצדורה. כלומר, כ"קופסה שחורה", אליה נכנס קלט, מתבצע FMM $(Fast\ Matrix\ Multication)$, ויוצא פלט. נשתמש כהנחה שכפל מטריצות עלותה $O(n^\omega)$ ונעזר בכך להוריד זמן ריצה של אלגוריתמים.

כעת נראה את האלגוריתם של סיידל, באשר $G = (V, E)$ הוא גרף לא מכוון ולא ממושקל, ונרצה לפתור את $APSP$ על G שיעשה בדיוק אותו דבר שתיארנו כאן ויגיע לסיבוכיות זמן של $O(|V|^\omega log(v))$. כמובן שאם בעתיד הערך של ω יקטן, גם הזמן של סיידל יקטן כי הוא משתמש בכפל מטריצות באופן ישיר.

0.2 האלגוריתם של $Seidel$

0.2.1 הגדרת הבעיה

קלט: גרף $G=(V,E)$ לא מכוון ולא ממושקל.
פלט: $APSP$.

נשתמש בהנחה מקלה - הגרף G קשיר. מכאן נקבל $|E|\geq|V|-1$. ניתן להניח זאת כי אם הגרף לא קשיר, נוכל לפתור את הבעיה עבור כל רכיב קשירות בנפרד (שכן מחשבים מרחק מסלול קצר ביותר בין קודקודים.)

כמה זמן לוקח להמיר גרף שמיוצג ע"י רשימת שכנויות לייצוג ע"י מטריצת שכנויות? $O(|V|^2)$.
נניח כי הגרף מיוצג ע"י מטריצת שכנויות. כדלקמן -

$$A_{u,v}:=\left\{\begin{array}{cc}1 & (u,v)\in E\\ 0 & else\end{array}\right\}$$

המטריצה A סימטרית, כיוון שG אינו מכוון ולכן $A_{u.v}=A_{v,u}$

0.2.2 כפל מטריצות בוליאני

בכפל מטריצות רגיל מתקיים לפי עדי בן צבי - $C_{ij}=\sum_{k=1}^{n}A_{ik}B_{kj}$.

בכפל מטריצות בוליאני (BMM) שנקרא $Boolean\ Metrix\ Multipcation$:

$$C_{ij}=\bigvee_{k=1}^{n}a_{ik}\wedge b_{kj}$$

כלומר, כאשר מייצגים מטריצות בוליאניות אנחנו מסתכלים על המיקום הij. הוא מכפלה של השורה הi במטריצה A והעמודה הj במטריצה B, מספיק שיהיה קיים אינדקס אחד k עבורו הערך הk בשורה הi והערך הk בעמודה הj הוא 1, אזי נגדיר $c_{ij}=1$. אחרת, $c_{ij}=0$.

נראה כי ניתן לחשב BMM באמצעות FMM: מדוע? במטריצות בוליאניות, נקבל ערך ששונה מאפס אמ"מ היה ערך k עבורו $a_{ik},b_{kj}\neq 0$. מכאן שאם יצא לנו ערך $c_{ij}\neq 0$ נגדירו כעת 1, ואם יצא 0 הוא ישאר אפס. עלות BMM תהיה $O(n^{\omega})$.

באלגוריתם של סיידל, אנחנו נשתמש במטריצה A שהינה מטריצת שכנויות, ונגדיר את המטריצה:

$$A'=A^2\vee A$$

באשר A^2 היא מטריצת השכנויות שמוכפלת בעצמה, בכפל בוליאני.

נרצה להבין מה המשמעות של A^2. מכפלה של מטריצת השכנויות עם עצמה: A^2_{ij} משמעות הדבר היא שלקחנו את השורה הi: כל השכנים של v_i, ולקחנו את השורה הj- כל השכנים של v_j, וחישבנו:

$$A^2_{ij}=\bigvee_{k=1}^{n}a_{ik}\wedge a_{kj}$$

כלומר ־ מסתכלים על כל השכנים של i, כל השכנים של j: אם מוצאים k מסויים שהוא שכן של שניהם, המשמעות היא שיש מסלול באורך 2 בין i לj.

טענה: במיקום $A_{i,j}^2 = 1$ אמ"מ קיים מסלול באורך 2 בין קודקוד i לקודקוד j.

המשמעות היא, שבגרף שמיוצג ע"י A^2 (נתייחס אליה גם כמטריצת שכנויות) יש קשת בין קודקוד i לקודקוד j אמ"מ יש מסלול באורך 2 בין i לj.

נזכר כי: רצינו להגדיר את המטריצה כך $A' = A^2 \vee A$. A^2 משמעות הדבר היא עבור כל המסלולים באורך 2 וA עבור קשתות המקוריות. כאשר נבצע or בין המטריצות, במטריצת הפלט A' יש 1 אמ"מ בין קודקוד i לj יש מסלול באורך 1 (קשת) או מסלול באורך 2.

טענה: $A'_{ij} = 1$ אמ"מ בG קיים מסלול באורך 1 או 2 בין הקודקודים i לj.

הערה חשובה: $A_{ii}^2 = 1$ אמ"מ לצומת יש שכן שכן משמעות הדבר כי קיים k עבורו $A_{ik} = 1$ וגם $A_{ki} = 1$. אם נסתכל על זוג הקודקודים $i - j$ עם הקשת בניהם, נראה כי המסלול $i \to j \to i$ הוא גם מסלול באורך 2. שחזר למקום. משמעות הדבר הינה, שA^2 על כל האלכסון יהיה אחדות. ומה זה אומר אם קיים מסלול באורך 1 מi לi? שישנה לולאה עצמית. ולכן ־ **תמיד אנחנו נאפס את האלכסון הראשי של A^2.**

0.2.3 טענות $1, 2$

הגדרה: נגדיר $\delta'(u,v)$ ־ אורך המסלול הקצר ביותר מu לv בG', כאשר G' הוא הגרף המיוצג ע"י A'.

טענה 1: $\delta'(u,v) = \left\lceil \frac{\delta(u,v)}{2} \right\rceil$

אינטואיציה לטענה: ראינו כי בגרף G' יש קשת בין קודקודים שG היה בניהם מסלול באורך 2, משמעות הדבר היא שעל כל צלע שהולכים בG' צריך ללכת פי 2 בגרף G. כלומר $x' = 2x$ ומכאן $x = \frac{x'}{2}$.

הוכחה: נסמן בP מסלול קצר ביותר מu לv בG.

אם האורך של p הוא $2k$ עבור $k \in \mathbb{N}$ אזי קיים מסלול מאורך k בין u לv בG', נסמנו P'. כיוון שהמסלול P' מדלג על כל קודקוד שני בP. כלומר אם $P = (v_0, ..., v_{2k-1})$ אזי $P' = (v_0, v_2, v_4, ..., v_{2k-2})$. עלינו להראות כי P' הוא מסלול קצר ביותר בין u לv. נב"ש כי קיים מסלול $\hat{p} = (v_0, ..., v_{m-1})$ באשר $m < k$. אזי המסלול $p'' = (v_0, u_0, v_1, u_1, ..., v_{m-1}u_{,m-1})$ קיים בG (כי בG' אנחנו יכולים לדלג על קודקוד, בG לא), כלומר מצאנו מסלול בין u לv בG שאורכו $2m < 2k$ בסתירה לכך שP הוא המסלול הקצר ביותר. מסקנה ־ P' מסלול קצר ביותר בין u לv בG'. ולכן אכן $\delta'(u,v) = k = \frac{2k}{2} = \frac{\delta(u,v)}{2}$

ההוכחה עבור המקרה האי זוגי ־ דומה מאוד.

מסקנה: אם למשל $\delta'(u,v) = 4$ אזי $\delta(u,v) = 7 \vee 8$. נרצה להבין מתי $\delta(u,v)$ יקבל איזה ערך. אם $\delta(u,v)$ זוגי הוא יקבל 8, ואם הוא אי זוגי הוא יקבל 7. **נרצה לפתח שיטה שתבדוק האם $\delta(u,v)$ הוא זוגי או אי זוגי.**

נסתכל על קודקוד w שהוא שכן של v. כמו כן, ישנו מסלול קצר ביותר בין u לv. וכן, קיים מסלול קצר ביותר בין u לw וכן לפי א"ש המשולש: $\delta(u,v) - 1 \leq \delta(u,w) \leq \delta(u,v) + 1$ (נשים לב כי באי השוויון השתמשנו פעמיים). מכאן נחלק למקרים:

א. **מקרה ראשון -** $[\delta(u,v)=\delta(u,w)]mod2$ **:** כלומר או ששניהם זוגיים, או ששניהם אי זוגיים.
במקרה זה אנו רואים שחייב להיות כי $\delta(u,v)=\delta(u,w)$. מדוע? מאי שוויון המשולש - ראינו כי ההפרש $|\delta(u,w)-\delta(u,v)|\leq 1$ ומכאן שההפרש שווה ל0 או 1. עם זאת, הזוגיות/אי זוגיות שלהם שווה ולכן לא יתכן שההפרש בניהם הוא אחד. מכאן ההפרש בניהם אפס - כלומר $\delta(u,v)=\delta(u,w)$. כמו כן, זה אומר כי w לא נמצא על המסלול הקצר ביותר מu לv. כמו כן, מכאן נקבל גם כי $\delta'(u,v)=\delta'(u,w)$.

ב. **מקרה שני -** $\delta(u,v)$ **זוגי ו**$\delta(u,w)$ **אי זוגי:**

$$\delta'(u,w)=\left\lceil\frac{\delta(u,w)}{2}\right\rceil=_{\delta(u,w)-odd}\frac{\delta(u,w)+1}{2}\geq_{\delta(u,v)-1\leq\delta(u,w)}\frac{\delta(u,v)-1+1}{2}=\frac{\delta(u,v)}{2}=\delta'(u,v)$$

כלומר - $\delta'(u,w)\geq\delta'(u,v)$

ג. **מקרה שלישי -** $\delta(u,v)$ **אי זוגי ו**$\delta(u,w)$ **זוגי:**

$$\delta'(u,v)=\left\lceil\frac{\delta(u,v)}{2}\right\rceil=_{\delta(u,v)-odd}\frac{\delta(u,v)+1}{2}\geq_{\delta(u,v)\geq\delta(u,w)-1}\frac{\delta(u,w)-1+1}{2}=\frac{\delta(u,w)}{2}=\delta'(u,w)$$

כלומר - $\delta'(u,v)\geq\delta'(u,w)$

מסקנה: אם $\delta(u,v)$ זוגי, אזי לכל שכן w של v מתקיים $\delta'(u,w)\geq\delta'(u,v)$
אם $\delta(u,v)$ אי זוגי, אזי לכל שכן w של v מתקיים $\delta'(u,w)\leq\delta'(u,v)$

מסקנה מהמסקנה - אם אנחנו יכולים לחשב את כל δ', אנו יודעים לבדוק יחס גדול־קטן בניהם. ומכאן: אנחנו יכולים להכריע האם δ זוגי או שאי זוגי. פרט לבעיה אחת - מה קורה אם $\delta'(u,w)=\delta'(u,v)$? לשם כך נצטרך להעזר בטענה הבאה.

אנו נמצאים במקרה של $\delta(u,v)$ אי זוגי ו$\delta(u,w)$ זוגי. יודעים כי $\delta'(u,v)\geq\delta'(u,w)$. נראה כי אם נסתכל על שכן מסויים מאוד w של v, נצליח להגיע במקרה זה למסקנה מעט אחרת. נסתכל על המסלול הקצר ביותר מu לv. נסמן בx את השכן של v על המסלול (זה שנמצא קודקוד אחד לפניו במסלול הקצר ביותר). מתכונות מסלולים קצרים ביותר (למה 1) מתקיים

$$\delta(u,v)=\delta(u,x)+1$$

שקול לחלוטין:

$$2k=\delta(u,x)=\delta(u,v)-1$$

נסמן את הביטוי הנ"ל ב$2k$. אכן $\delta(u,x)$ זוגי כי $w=x$ במקרה זה. מכאן $\delta(u,v)=2k+1$.
לפי ההגדרה:

$$\delta'(u,x)=\left\lceil\frac{\delta(u,x)}{2}\right\rceil=\left\lceil\frac{2k}{2}\right\rceil=\lceil k\rceil=k$$

$$\delta'(u,v) = \left\lceil \frac{\delta(u,v)}{2} \right\rceil = \left\lceil \frac{2k+1}{2} \right\rceil = \left\lceil k + \frac{1}{2} \right\rceil = k+1$$

מכאן ניתן לראות כי $\delta'(u,x) < \delta'(u,v)$. כלומר - בהינתן שנדע למצוא את השכן של v על המסלול הקצר ביותר מu לv. נוכל לדעת כי במקרה בו $\delta(u,v)$ אי זוגי ו$\delta(u,w)$ זוגי אם נסתכל על אותו שכן כזה x, יתקיים $\delta'(u,x) < \delta'(u,v)$ (ולא יתקיים שוויון בניהם), נשתמש בזה בהוכחה הבאה.

הרעיון של סיידל היה להסתכל על כל השכנים יחד של v.

טענה 2: $\delta(u,v)$ זוגי $\iff \sum_{w\in\Gamma(v)} \delta'(u,w) \geq deg(v) \times \delta'(u,v)$

(מדוע אנחנו זקוקים לטענה זו? אנו יודעים להכריע אודות הדרגה. אם נצליח לחשב (ונצליח, באופן רקורסיבי) את הΣ הנ"ל, נדע להכריע האם $\delta(u,v)$ הינו זוגי. אם לא - הוא בהכרח אי זוגי.)

הוכחה:

$\Leftarrow$ אם $\delta(u,v)$ זוגי - אזי לכל שכן w של v יתקיים $\delta'(u,w) \geq \delta'(u,v)$ (כפי שראינו קודם לכן), ולכן

$$\sum_{w\in\Gamma(v)} \delta'(u,w) \geq \sum_{w\in\Gamma(v)} \delta'(u,v) = deg(v)\delta'(u,v)$$

$\Longrightarrow$ נראה קונטרה פוזיטיב. נניח $\delta(u,v)$ אי זוגי. אזי כפי שראינו מעלה במקרה זה לv יש שכן x שעבורו $\delta'(u,x) < \delta'(u,v)$ וכן לכל שכן w של v מתקיים $\delta'(u,w) \leq \delta'(u,v)$

$$\sum_{w\in\Gamma(v)} \delta'(u,w) = \delta'(u,x) + \sum_{w\in\Gamma(v)\setminus x} \delta'(u,w)$$

החלק הימני של הביטוי $\geq (deg(v)-1) \times \delta'(u,v)$ וכן החלק השמאלי $< \delta'(u,v)$ ולכן נקבל

$$\sum_{w\in\Gamma(v)} \delta'(u,w) = \delta'(u,x) + \sum_{w\in\Gamma(v)\setminus x} \delta'(u,w) < \delta'(u,v)deg(v)$$

כפי שרצינו להראות.

0.2.4 האלגוריתם

עוד לפני שנדון באלגוריתם - נרצה לראות אלגוריתם גנרי:

ALG1(A)

if A is all 1s except for the diagonal

  return A

else $\Delta' \leftarrow \mathrm{ALG1}(A^2 \vee A)$

  for $u, v \in V$

    if $\delta(u,v)$ is odd

      $\delta(u,v) = 2\delta'(u,v) - 1$

    else $\delta(u,v) = 2\delta'(u,v)$

  return Δ

האלגוריתם של סיידל יתבסס על אלגוריתם גנרי זה. אלגוריתם זה פשוט יותר להבנה ־ האלגוריתם מקבל את מטריצת השכנויות A. האלגוריתם יחזיר מטריצה Δ באשר:

$$\Delta_{i,j} = (\delta(i,j))$$

ראשית, ישנו תנאי עצירה: אם G הוא קליקה, נרצה להחזיר את A. (מדוע? בקליקה לכל $u \neq v \in V$ מתקיים $\delta(u,v) = 1$, במקרה זה מטריצת השכנויות של הקליקה הינה 0 באלכסון ובכל שאר המקומות 1. במצב זה ־ זה בדיוק מטריצת ה$APSP$ שנרצה להחזיר).

אחרת, אנחנו נרצה לגשת שוב לאלגוריתם עם $A' = A^2 \vee A$. ערך זה, יכנס לΔ'. נשים לב כי זה יתבצע בתחילת האלגוריתם שוב ושוב, עד שנקבל $(\Delta')^n$ עבור n כלשהו, באשר הוא קליקה.

לאחר מכן, נעבור על כל זוג קודקודים בV. אם $\delta(u,v)$ הוא אי זוגי, אזי $\delta(u,v) = 2\delta'(u,v) - 1$. אחרת, $\delta(u,v) = 2\delta'(u,v)$ [נובע ישירות מטענה 1 שראינו], לבסוף מחזירים את Δ. הרעיון של סיידל התבסס על כיצד אנחנו מכריעים אודות הזוגיות של $\delta(u,v)$ בשביל לבצע מה שעשינו באלגוריתם הגנרי.

כעת, נראה את האלגוריתם של סיידל:

```
SEIDEL(A)
1   if A is all 1s except for the diagonal
2       return A
3   else Δ' ← SEIDEL(A² ∨ A)
4       M ← Δ' · A
5       for u, v ∈ V
6           if m_{u,v} < deg(v)δ'(u, v)
7               δ(u, v) = 2δ'(u, v) − 1
8           else δ(u, v) = 2δ'(u, v)
9       return Δ
```

האלגוריתם זהה בהתחלה לגנרי, ובשורה 6 אנחנו משתמשים בטענה 2: אם $m_{u,v} < deg(v)\delta'(u,v)$ אזי אנחנו במקרה האי זוגי. מכאן, ישירות לפי טענה 2 אפשר להבין לבד מהו $m_{u,v}$: $\sum_{w\in\Gamma(v)} \delta'(u,w)$.

בשורה 4 אנחנו מגדירים את M: היא מכפלה של מטריצת השכנויות המקורית A עם המטריצה Δ' - נראה כי במיקום הu,v ישנה מכפלה של כל השורה הu בΔ' עם העמודה הv בA. נראה כי כיוון שהמכפלה בוליאנית, המכפלה של העמודה והשורה יתנו לנו את סכום כל הδ' של הקודקודים שהם שכנים של v. מדוע? השורה הu היא שורה שמחזיקה את האיברים $(\delta'(u,v_1),....,\delta'(u,v_n))$, מכאן שערך שכזה יכנס למכפלה אמ"מ v_i הוא שכן של v. מכאן שהמכפלה הנ"ל תניב את $\sum_{w\in\Gamma(v)} \delta'(u,w)$. כלומר,

$$m_{u,v} = \sum_{w\in\Gamma(v)} \delta'(u,w)$$

ומכאן קיבלנו את האלגוריתם שמבוסס ישירות על טענה 2 - אם $\sum_{w\in\Gamma(v)} \delta'(u,w) < deg(v)\delta'(u,v)$ המשמעות היא כי $\delta(u,v)$ אי זוגי ולכן $\delta(u,v) = 2\delta'(u,v) - 1$. אחרת, $\delta(u,v)$ זוגי ולכן $\delta(u,v) = 2\delta'(u,v)$.

סיבוכיות זמן ריצה:

נשים לב כי ניתן לחשב את דרגות הקודקודים ב$O(|V|^2)$ זמן באופן טריוואלי - ניצור מערך של קודקודים, עבור כל קודקוד $v \in V$ נעבור על השורה המתאימה במטריצת השכנויות ונסכום את מס' הקודקודים u שקיימת $e = (v,u)$ בניהם. נעבור על $O(|V|)$ קודקודים ובכל פעם שכזו נעבור על $O(|V|)$ קודקודים וסה"כ זמן הריצה יהיה $O(|V|^2)$.

מכאן, שבבירור ניתן לחשב את שורה 6 באלגוריתם. נראה כי שורות $4-8$ באלגוריתם עולות $O(|V|^\omega)$ זמן, כיוון שמבצעים מעבר על $|V|$ קודקודים ובהם מבצעים פעולות שהינם $O(1)$ וכן כופלים בBMM שתי מטריצות, מה שעולה $O(|V|^\omega)$ זמן. סה"כ $|V| + |V|^\omega = O(|V|^\omega)$ כי $\omega \geq 2 > 1$.

מהי נוסחת הנסיגה של האלגוריתם? מתי נגיע לתנאי העצירה?

הבחנה: אם אורך המסלול הקצר ביותר בG הוא ℓ אזי בG' אורך המסלול הקצר ביותר מu לv הוא $\lceil \frac{\ell}{2} \rceil$. מכאן, שלאחר $log(|V|)$ קריאות רקורסיביות אורך המסלול הקצר ביותר הגרף הוא

קליקה. מדוע? אורך המסלול הקצר ביותר הוא לכל היותר $|V|-1$ קשתות. אנחנו נרצה לדעת מתי נגיע לקליקה - כלומר מתי אורך המסלול הקצר ביותר יהיה 1. מכאן ש

$$\frac{|V|-1}{2^n} = 1 \Longleftrightarrow |V|-1 = 2^n \Longleftrightarrow n = log(|V|-1) = O(log|V|)$$

וסה"כ נקבל כי סיבוכיות זמן הריצה היא מס' האיטרציות $O(log|V|)$ כפול הזמן בכל אטרציה $O(|V|^\omega)$ ונקבל -

$$O(|V|^\omega log|V|) < O(|V|^3)$$

הבחנה. אם ידוע כי האורך הכי גדול של מסלול בגרף בין שני קודקודים הוא d, אזי סיבוכיות זמן הריצה של האלגוריתם הינה $O(|V|^\omega log(d))$.

הבחנה. גם אם נדע אלגוריתם טוב יותר לBMM, בכל מקרה בכל איטרציה מחשבים את M שהיא לא כפל מטריצות בוליאניות, ולכן בכל מקרה זה זמן הריצה.

0.3 A^*

נתון גרף מכוון וממושקל $G=(V,E)$ וכן שני קודקודים $s,t \in V$. נרצה למצוב מק"ב מs לt. נניח כי $w: E \to \mathbb{R}^+$.

דייקסטרה ירוץ כאן בזמן $O(|V|log|V| + |E|)$.

נרצה לפתור כרגע בעיה בין $source$ ל$target$: רק בין s לt. נשים לב כי טענו שלא קיים פתרון לבעיה זו של מקור יחיד בזמן יותר טוב מ$SSSP$. אך אנחנו לא נדבר על המקרה הגרוע ביותר.

איך נעזר בדייקסטרה? נרצה לעצור באשר t יוצא מהתור כי אז הוכחנו שמצאנו מק"ב מs אל t - ואין לנו צורך בהמשך הריצה של דייקסטרה.

כעת נדגיש: הרעיון של A^* הינו רעיון אלגוריתמי וניתן להגדיר אלגוריתמיים שפועלים לפי טכניקה A^*, אנחנו לא נראה אלגוריתם ישיר שפותר A^* אלא כיצד משתמשים בA^* להגדרת בעיות ופתרונם.

ננסה לחשוב על הכיוון הבא - נגדיר את פונקציית הפוטנציאל $P: V \to \mathbb{R}$. מטרת הפונקציה היא כשצומת v קרוב אל t אזי P קטן יותר. ואז, מה שנעשה באלגוריתם של דייקסטרה יהיה להוסיף לכל $d[v]$ את $p(v)$ כלומר $d[v] = d[v] + p(v)$, וכך אנחנו בתחילת הריצה של דייקסטרה נתעדף את הצמתים שקרובים אל t. נשים לב - כל התוספת הנ"ל פוגעת בכל הנכונות של דייקסטרה. לכן - לא נשתמש בזה.

נגדיר פונקציית משקל חדשה:

$$w'(u,v) = w(u,v) + p(v) - p(u)$$

נראה כי

$$w'(P_{u\to v}) = w(P_{u\to v}) + p(v) - p(u)$$

ולכן כל המסלולים בין u לv עם אותה התוספת $p(v)-p(u)$ ולכן הסדר ביניהם נשאר, ובפרט למק"ב.

עם זאת, יש המון בעיות שאנו זקוקים לטפל בהם - בין צמתים שונים (למשל u לש v וu לש w) לא מובטח שנשמר הסדר. כמו כן, בשביל שדייקסטרה יעבור נרצה כי $w'(u,v) \geq 0$.

הגדרה: אם לאחר ההתמרה (הגדרת w') כל $w' \geq 0$ אזי p (פונקציית הפוטנציאל) הינה פיזיבילית. כלומר לכל $(u,v) \in E$ $w'(u,v) \geq 0 \iff p$ פיזיבילית.

למה 1: אם p פיזיבילית, ויהי $t \in V$ כך ש $p(t) \leq 0$, אזי $\forall v \in V$: $p(v) \leq \delta(v,t)$

הוכחה: יהי מסלול מv אל t.

$$0 \leq w'(P_{v\to t}) = w(P_{v\to t}) + p(t) - p(v) \leq w(P_{v\to t}) - p(v)$$

וסה"כ נקבל $p(v) \leq w(P_{v\to t})$. בפרט, זה נכון עבור המסלול הקצר ביותר כלומר $p(v) \leq \delta(v,t)$

מה קורה באשר $p(v) = \delta(v,t)$ לכל קודקוד?

$$p_{s\to t} = (s, v_1, ..., v_k, t)$$

נסתכל על (v_i, v_{i+1}) במסלול.

$$w'(v_i, v_{i+1}) = w(v_i, v_{i+1}) + \delta(v_{i+1}, t) - \delta(v_i, t) = \delta(v_i, t) - \delta(v_i, t) = 0$$

משום ש$w(v_i, v_{i+1}) + \delta(v_{i+1}, t) = \delta(v_i, t)$. ולכן - כל קשת במק"ב מ$s \to t$ תהיה במשקל 0. נשים לב - כיוון שכל המשקלים על המסלול הנ"ל הינם אפס, דייקסטרה ישר ירוץ על המסלול שלנו מs לt. הוא ראשית יוציא את s עם $d[s] = 0$ ואז יבצע סדרת הקלות קודם כל על המק"ב שלנו מs לt. מכאן, שהשאיפה שלנו היא למצוא פונקציית פוטנציאל שמקרבת את $p(v)$ כמה שיותר אל $\delta(v,t)$. (שכן אנחנו לא יודעים את $\delta(v,t)$ שכן בשביל לדעת אותו צריך להריץ דייקסטרה, ואנחנו לא מעוניינים לעשות זאת. נשים לב כי את $\delta(v,t)$ נוכל למצוא אם נריץ דייקסטרה על G^T מt.)

מסקנה מהחישוב: אם נמצא ערך $p(v)$ קרוב מאוד ל$\delta(v,t)$ אזי ערך הקשת יהיה קרוב יותר לאפס.

למה 2: אם p פיזיבלי ו$p(t) > 0$ אזי ניתן להפוך לp' פיזיבלי עם $p'(t) \leq 0$.
הוכחה:
נגדיר

$$p'(v) = p(v) - p(t)$$

ברור כי עבור $p'(t) = 0$. כל שניתן להראות הוא כי p' פיזיבילי. תהי $(u,v) \in E$. אזי,

$$w''(u,v) = w(u,v) + p'(v) - p'(u) = w(u,v) + p(v) - p(t) - p(u) + p(t) = w'(u,v) \geq 0$$

שכן מראש הנחנו כי p פיזיבלי ולכן $w'(u,v) \geq 0$.

דוגמה ראשונה לשימוש בA^*:
נרצה לבדוק מרחק קצר ביותר בין שני צמתים במישור. אם ניתן לשכן צמתים במישור, נסתכל על מפה דו מימדית וצומת (x_1, y_1) ו(x_2, y_2). נסמן $||g(u) - g(v)||$ כמרחק האוקלידי בין הקודקודים והוא $\sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$.
אם נסתכל על פונקציית פוטנציאל $p(v) = ||g(v)-g(t)||$ אזי בהכרח $p(t) = ||g(t)-g(t)|| = 0$.
מכאן $p(t) \geq 0$ ונראה כי היא פיזיבילית -

$$w'(u,v) = w(u,v) + p(v) - p(u) = w(u,v) + ||g(v) - g(t)|| - ||g(u) - g(t)||$$

נראה כי אם נדרוש $w(u,v) \geq ||g(u)-g(v)||$ (**ברור שהכביש יהיה ארוך יותר מהמרחק האווירי**).
נניח שזה מתקיים, אזי

$$\geq ||g(u) - g(v)|| + ||g(v) - g(t)|| - ||g(u) - g(t)|| \geq 0$$

באשר המעבר האחרון מגיע מאי שוויון המשולש על מרחקים אוקלידים.

דוגמה שנייה לשימוש בA^*: נניח צומת מרכזי f וידוע $\delta(v,f)$ מכל $v \in V$ אליו.)כלומר כמה זמן לוקח לי להגיע מכל קודקוד ליעד מרכזי(. נראה כי אני כן מבזבז נתונים ומפעיל דייקסטרה אל f מכל קודקוד אך רוצה להשתמש בנתון זה לדעת יותר - למשל, בהינתן המידע הזה, לאיזה צומת אני מעוניין להגיע מהמיקום הנוכחי שלי, כך שסה"כ המרחק ממני לf יהיה קצר ביותר.
בהינתן מידע זה נרצה להגדיר את הפונקציה הבאה:

$$p(v) = \delta(v,f) - \delta(t,f)$$

אכן $p(t) = 0$. נוכיח פיזיבלי:

$$w'(u,v) = w(u,v) + p(v) - p(u) = w(u,v) + \delta(v,f) - \delta(t,f) + \delta(t,f) - \delta(u,f) =$$

$$w(u,v) + \delta(v,f) - \delta(u,f) \geq 0$$

שכן המעבר נובע אוטומטית מאי שוויון המשולש.

מסקנה: ניתן להריץ את האלגוריתם של דייקסטרה אם מגדירים לו פונקציית פוטנציאל טובה, פיזיבילית, ולקבל שבאופן ישיר האלגוריתם ראשית יבצע את המסלול שאני מעוניין בו, מה שיוריד את זמן הריצה.