31/12 בס"ד

37

נושא 9 - בעיות מסלולים קצרים

א) הגדרת הבעיה

בהינתן גרף ממושקל $G=(V,E)$, נרצה למצוא אלגוריתם שיחשב את המסלול הקצר ביותר בין כל שני צמתים בגרף. הערך המוחזר מהאלגוריתם יהיה מטריצה D בגודל $n \times n$, כך שבכל תא $D[i,j]$ במטריצה זו יהיה המרחק הקצר ביותר מ-i ל-j. $D[i,j]=\delta(i,j)$.

נניח כי המשקלים של כל קשת גם הם נתונים במטריצה W בגודל $n \times n$. בכל תא $W[i,j]$ אם קיים קשת מ-i ל-j אזי בתא זה יהיה משקל הקשת. אם אין קשת בין i ל-j יהיה בתא זה ∞, ואם $i=j$, כלומר הם אותה צומת יהיה בתא זה 0.

$$W[i,j]=\begin{cases}0 & i=j\\ w(i,j) & i\neq j\ (i,j)\in E\\ \infty & i\neq j\ (i,j)\notin E\end{cases}$$

נניח לעת עתה כי ב-G אין מעגלים שליליים, נחזור לכך בסוף הפרק.

ננסה מספר שיטות למציאת אלגוריתם יעיל לבעיה זו:

ב) נסיון ראשון - אלגוריתם בלמן-פורד/דייקסטרה

עבור כל צומת בגרף נפעיל אלגוריתם למציאת מסלול הכי קצר ממנו לכל שאר הצמתים. בגרף עם משקלים אי-שליליים נשתמש באלגוריתם דייקסטרה, ובגרף עם משקלים שליליים נשתמש באלגוריתם בלמן-פורד. זמן הריצה הוא:

אלגוריתם דייקסטרה - $O(n \cdot n^2)=O(n^3)$ בגרף צפוף, ו-$O(n^2 \log n)$ בגרף דליל. $O(n^2)$ צפוף / $O(n\log n)$ דליל

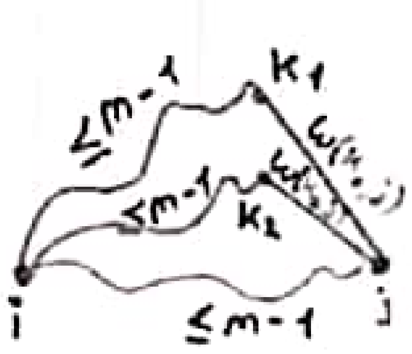
אלגוריתם בלמן-פורד - $O(m \cdot n^2)$. נקבל $O(n^4)$ בגרף צפוף ו-$O(n^3)$ בגרף דליל. $O(m \cdot n)$

ג) נסיון שני - תכנון דינמי

עבור כל צומת i נבנה מטריצה d_i בגודל $n \times n$. בכל תא (m,j) במטריצה d_i, אותה נסמן $d_{i,j}^{(m)}$, תחזיק את ערך המסלול הכי קצר מ-i ל-j כאשר ניתן להשתמש רק ב-m או פחות צלעות. מטרתנו בסוף היא לחשב את $d_{i,j}^{(n-1)}$ לכל $j \in V$, וככה לקבל את המסלול הכי קצר מצומת i לכל שאר הצמתים בגרף. נשים לב כי $d_{i,i}^{(m)}$ יהיה שווה 0 לכל $1 \le m \le n-1$, וכי $d_{i,j}^{(0)}=\infty$ לכל $j \neq i$. עבור כל שאר התאים נבנה את הנוסחה הרקורסיבית הבאה:

	1	2	...	i	...	n
0	∞	∞	∞	0	∞	∞
2				0		
3				0		
⋮				0		
				0		
n-1	?	?	?	0	?	?

$$d_{i,j}^{(m)}=\min\left\{d_{i,j}^{(m-1)},\ \min_k\left\{d_{i,k}^{(m-1)}+w(k,j)\right\}\right\}=\min_k\left\{d_{i,k}^{(m-1)}+w(k,j)\right\}$$



נבדוק לכל $k \in V$.

נמלא שורה אחר שורה בטבלה, ובכל תא נחשב את המינימום בין כל התאים בשורה מעל.

השוויון השני נובע עבור $k=j$: $d_{i,j}^{(m-1)}=d_{i,j}^{(m-1)}+w(j,j)$.

31/12 בס"ד

38

לאחר שנעבור על כל התא'ם במטריצה, נקבל בשורה האחרונה את המרחק הכי קצר מ-i אל כל צומת אחרת בגרף. ניקח שורה זו ונכניס אותה לשורה i במטריצה D.

בשיטה זו עבור כל צומת אנו בונים מטריצה בגודל $n \times n$ עם n^2 תא'ם, ובכל תא אנו צריכים לחשב n תא'ם. לכן סיבוכיות האלגוריתם היא: $O(n\cdot n^2\cdot n)=\underline{O(n^4)}$

3) ניסיון שלישי - כפל מטריצות

נגדיר את המטריצה $D^{(m)}$ שהיא מטריצה בגודל $n \times n$ ומורכבת מהכפלת מטריצת המשקלים W m פעמים בעצמה. $D^{(1)}=W,\ D^{(2)}=W\cdot W,\ D^{(m)}=\prod_{i=1}^{m} W$.

כפל המטריצות בשיטה זו הוא אינו בשיטה הרגילה. בהכפלה רגילה בין מטריצות $C=A\cdot B$ כל תא C_{ij} במטריצה C מתקבל מסכום כל המכפלות של שורה i במטריצה A לעמודה j במטריצה B, כלומר $C_{ij}=\sum_K A_{iK}\cdot B_{Kj}$. אמנם בשיטה הנ"ל נגדיר הכפלת מטריצות בדרך הבאה: כל תא C_{ij} במטריצה C מתקבל מהערך המינימלי בין כל החיבורים של שורה i במטריצה A לעמודה j במטריצה B, כלומר $C_{ij}=\min_K\{A_{iK}+B_{Kj}\}$.

$$\begin{array}{cc} A & B \\ \begin{array}{c|cccc} & x & x & x & x \\ i & 1 & 3 & 4 & 2 \\ & x & x & x & x \\ & x & x & x & x \end{array} \cdot & \begin{array}{cccc} & j & & \\ x & 2 & x & x \\ x & 5 & x & x \\ x & 6 & x & x \\ x & 7 & x & x \end{array} \end{array}$$

$C_{ij}=\min\{\underline{1+2},3+5,4+6,2+7\}$

לא נוכיח זאת אך במטריצה $D^{(m)}$ המתקבלת בשיטה זו, כל תא d_{ij} יחזיק את המסלול הקצר ביותר מ-i ל-j כאשר ניתן להשתמש רק ב-m קשתות. מטרתנו אם כן היא לחשב ולהחזיר את מטריצה $D^{(n-1)}$, שהרי המסלול הפשוט הארוך ביותר מכיל n-1 קשתות.

כדי לחשב את $D^{(n-1)}$ אם נכפיל את $D^{(m)}$ כל פעם ב-W, נצטרך n-1 הכפלות. בכל הכפלה נחשב n^2 ערכים, ועבור כל ערך נצטרך לבדוק n חיבורים. לכן הסיבוכיות תהיה $O(n^4)$. נוכל לשפר זאת אם נשתמש במטריצות קודמות שכבר חישבנו.

נשים לב כי את המטריצות שהמעלה שלהם היא חזקה של 2 $\underbrace{\overbrace{\overbrace{W\cdot W}^{W^2}\cdot W^2}^{W^4}\cdot W^4}_{W^8}, W^8\dots$ ניתן לחשב בקלות. מכיוון שבכפל איברים דומים מחברים חזקות, אם נציג את n-1 בייצוג בינארי, נדע איזה מטריצות בחזקה של 2 אנו צריכים להכפיל. לפי מיקום האחדות במספר הבינארי. במקרה הגרוע נקבל מספר בינארי שכולו אחדות, ואז מספר ההכפלות יהיה כאורך המספר הבינארי של n-1, שהוא $\log(n-1)$. לכן בדרך זו הסיבוכיות תהיה $O(n^3\log n)$.

$\underset{3210}{1011}=2^3+2^1+2^0=11$

$\underset{43210}{11001}=2^4+2^3+2^0=25$

$\underset{43210}{10101}=2^4+2^2+2^0=21$

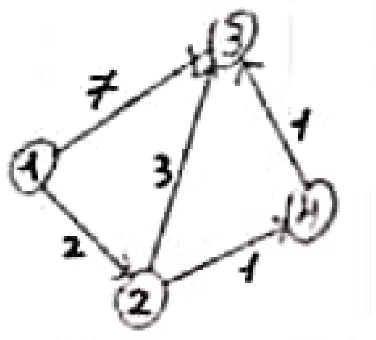
31/12

39

ה) נסיון רביעי - אלגוריתם פלויד וורשל (Floyd-Warshall)

באלגוריתם זה נמספר את כל הצמתים מ-1 עד n. $V=\{1,2,\dots,n\}$.

נסמן $d_{ij}^{(k)}$ - משקל המסלול הקצר ביותר מ-i ל-j כאשר $1,2,\dots,k$ יכולים להיות צמתי ביניים במסלול זה, כלומר שהמסלול יכול לעבור (אך לא חייב) בצמתים $1,2,\dots,k$, אך בשאר הצמתים $k+1,\dots,n$ הוא לא יכול לעבור. ככל שיש יותר צמתי ביניים כך מתרבים המסלולים האפשריים, $d_{ij}^{(k)}$ לוקח את הקצר מביניהם. נשים לב כי $d_{ij}^{(0)}\neq\infty$ רק אם יש קשת מ-i ל-j. ואז $d_{ij}^{(0)}=W_{ij}$



$d_{13}^{(0)}=7$ $d_{13}^{(3)}=1$
$d_{13}^{(1)}=7$ $d_{13}^{(4)}=1$
$d_{13}^{(2)}=5$

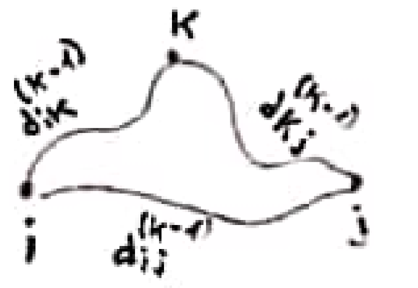
עבור כל שני צמתים i,j נרצה לחשב את $d_{ij}^{(n)}$, שהוא המסלול הקצר ביותר היכול לעבור בכל הצמתים, כלומר $d_{ij}^{(n)}=\delta(i,j)$.

נסמן $D^{(k)}$ מטריצה בגודל $n\times n$ שבה בכל תא i,j מוחזק הערך $d_{ij}^{(k)}$. כלומר מטריצה $D^{(k)}$ מחזיקה את משקל המסלולים הקצרים ביותר בין כל שני צמתים, כאשר רק $1,2,\dots,k$ הם צמתי ביניים במסלולים. נשים לב כי $D^{(0)}=W$, וכי מטרת האלגוריתם היא להחזיר את $D^{(n)}$.

אלגוריתם פלויד וורשל אומר שלשם כך יש לחשב את כל המטריצות $D^{(0)}, D^{(1)}, D^{(2)},\dots,D^{(n)}$, באמצעות הנוסחה הרקורסיבית הבאה:

$$\begin{cases} d_{ij}^{(0)}=W. \\ d_{ij}^{(k)}=\min\{d_{ij}^{(k-1)},\ d_{ik}^{(k-1)}+d_{kj}^{(k-1)}\} \end{cases}$$

כלומר המסלול הקצר ביותר מ-i ל-j או שלא עובר בצומת k או שכן עובר. נעבור בלולאה מ-1 עד n. בכל לולאה k נמלא את מטריצה $D^{(k)}$, על ידי שנעבור על כל התאים, שורה אחר שורה, ונשתמש בנוסחה הרקורסיבית כדי לחשב כל תא. הנוסחה הרקורסיבית תסתכל בערכים מהמטריצה הקודמת שמילאנו $D^{(k-1)}$. אלגוריתם זה יעבוד רק כאשר אין מעגלים שליליים בגרף!!



פסאודו קוד:

```
Floyd-Warshall (G,W) {
    D^(0) = W
    For (k=1 to n) {                 // כל מטריצה
        For (i=1 to n) {             // כל שורה במטריצה
            For (j=1 to n) {         // כל עמודה בשורה
                d_ij^(k) = min{d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1)}
            }
        }
    }
    return D^(n)
}
```

זמן הריצה של אלגוריתם זה הוא כמובן $\Theta(n^3)$.

31/12

1) סיכום זוגות מסלולים הקצרים ביותר

לפי סוג המשקל של הגרף נחליט איזה אלגוריתם להשתמש. האלגוריתמים היעילים ביותר לכל מקרה מסוכמים בטבלה.

זמן ריצה	אלגוריתם	סוג הגרף
$O(n(m+n))$	n פעמים BFS	לא ממושקל
$O(n^3)$ בגרף צפוף / $O(n^2 \log n)$ בגרף דליל	n פעמים דייקסטרה	ממושקל אי-שליליים
$O(n^3)$	פלויד - וורשל	ממושקל ללא מעגלים שליליים
$O(n^2 \cdot m)$	n פעמים בלמן - פורד	ממושקל כללי

2) משקול מחדש - אלגוריתם ג'ונסון (Johnson)

בטבלה לעיל ניתן לראות שיש יתרון משמעותי לאלגוריתם דייקסטרה על פני בלמן פורד, וכן יש יתרון משמעותי לדייקסטרה על פני פלויד-וורשל בגרף דליל.
הבעיה היא שלא ניתן להפעיל דייקסטרה על משקלים שליליים. פתרון לבעיה זו הוא לבצע משקול מחדש לגרף, כך שלאחר פעולה זו נפעיל אלגוריתם דייקסטרה.
המשקול מחדש צריך לשמור על שתי תכונות:
(1) לאחר המשקול מחדש משקל כל הקשתות יהיו אי-שליליים, $\hat{w}(u,v)\geq 0$.
(2) המסלול הקצר ביותר בין כל שני קודקודים בגרף המקורי ובגרף המתקבל לאחר המשקול זהים. אין הכוונה לגודל המרחק אלא בדרך הצמתים דרכם עוברים המסלול.

<u>אלגוריתם ג'ונסון:</u> נמשקל כל קשת בגרף לפי הנוסחה: $\hat{w}(u,v)=w(u,v)+h(u)-h(v)$
כאשר $\hat{w}(u,v)$ הוא המשקל החדש של הקשת $(u,v)\in E$, $w(u,v)$ הוא משקל הקשת המקורי, ו-h היא פונקציה המקבלת צומת ומחזירה מספר ממשי $h:V\to\mathbb{R}$, המוגדרת בדרך הבא: נוסיף לגרף צומת נוספת s ונמתח קשתות לכל צמתי הגרף. משקל כל קשת כזו הוא 0. נחשב את המסלול הקצר ביותר מ-s לכל אחד מצמתי הגרף באמצעות בלמן-פורד (שהרי המשקלים עדיין שליליים). כעת נגדיר עבור כל צומת $v\in V$ את $h(v)$ להיות המסלול הקצר ביותר מ-s ל-v, $h(v)=\delta(s,v)$.
נוכיח כי אכן משקול זה עונה על שתי התכונות לעיל:

(1) $\underbrace{\delta(s,v)\leq\delta(s,u)+w(u,v)}_{\text{אי-שוויון המשולש}} \Rightarrow \underbrace{h(v)\leq h(u)+w(u,v)}_{\text{מהגדרת } h} \Rightarrow \underbrace{w(u,v)+h(u)-h(v)\geq 0}_{\text{העברת אגפים}} \Rightarrow \underbrace{\hat{w}(u,v)\geq 0}_{\text{מהגדרת המשקל}}$

(2) עבור כל מסלול $P=\langle v_1,\dots,v_k\rangle$ נקבל כי: $\hat{w}(P)=\hat{w}(v_1,v_2)+\hat{w}(v_2,v_3)+\dots+\hat{w}(v_{k-1},v_k)$
$=w(v_1,v_2)+h(v_1)-\cancel{h(v_2)}+w(v_2,v_3)+\cancel{h(v_2)}-\cancel{h(v_3)}+\dots+w(v_{k-1},v_k)+\cancel{h(v_{k-1})}-h(v_k)=w(P)+h(v_1)-h(v_k)$

נניח ש-P המסלול הכי קצר בגרף המקורי, ונניח בשלילה שאינו הכי קצר בגרף החדש. לכן:
$\underbrace{\hat{w}(P')=w(P')+h(v_1)-h(v_k)}_{\text{שורה למעלה}} \underbrace{<}_{\text{הנחה בשלילה}} \underbrace{\hat{w}(P)=w(P)+h(v_1)-h(v_k)}_{\text{שורה למעלה}} \underbrace{\Rightarrow}_{\text{העברת אגפים}} w(P')<w(P)$!? בסתירה למינימליות של P

שלבי האלגוריתם:
נוסיף צומת s והשתות במשקל 0 לכל הצמתים.
בלמן-פורד מ-s.
נמשקל מחדש באמצעות הנוסחה.
נריץ דייקסטרה n פעמים.
נשיר את המרחקים הקצרים שהתקבלו חזרה למשקלים העיקריים:
$D[u,v]=\hat{\delta}(u,v)-h(u)+h(v)$
נוסחת המרה חשובה!! נחזיר את D.