תכנון וניתוח אלגוריתמים
הרצאה 20

מסלולים קצרים לפי
דייקסטרה

- בסעיף זה נתוודע אל הבעיה השימושית - הנקראת בעיית מסלולים קצרים ונציג לה פתרונות המשתמשים ברשתות (גרף ממושקל).
- נציג פתרונות שונים ונחקור את סיבוכיות זמן הריצה שלהם.
- **<u>הגדרה פורמלית של בעיית המסלול הקצר</u>**
- נתון גרף משוקלל $G=(V,E)$.

- לכל קשת מיוחסת מספר אשר יכול לייצג מחיר, מרחק בין שני ישובים, עלות בניית כביש שיקשר בין שני הישובים , ממוצע מספר הלקוחות העוברים מטרמינל אחד לטרמינל אחר , זמן בכדי להגיע מישוב אחד לישוב אחר ועוד.
- בלי הגבלת הכלליות נניח שמספר שמיוחס לקשת יציין את המרחק בין שני ישובים.

- כל קודקוד ברשת מייצג ישוב וכל קשת ברשת מייצגת כביש בין שני ישובים והמספר שעל הקשת מייצג את האורך של כביש זה (מרחק בין שני ישובים).
- **הבעיה היא** – מהו אורך המסלול הקצר ביותר מקודקוד מקור
- (ישוב מסויים)לקודקוד אחר (לישוב אחר) ברשת, בהתחשב למרחקים שישנם בין הישובים השונים.

- כאמור בעיות רבות דומות באופיין לבעיה זו .
- למשל, המספר שמיוחס לקשת מייצג עלות לבניית כביש בין שני ישובים כלשהם
- במקרה זה לא ניתן לבנות כבישים ישירים מכל ישוב לכל ישוב אחר
- מוטלת עלינו לבנות במינימום עלות את הישובים כך שתהיה אפשרות להגיע מכל ישוב לכל ישוב אחר.

- להלן התיאור הפורמלי של הבעיה:
- נתון גרף קשיר $G=(V,E)$
- עם קדקודים הממוספרים באופן אקראי מ- 0 עד n-1 , כלומר $|V|=n$
- עם פונקציית משקל $W: E \rightarrow R$, אשר מייחסת לכל קשת מספר שנכנה אותו בשם מרחק ,ועם המרחקים E_{ij} לכל קשת (i,j) המחברת שני קדקודי הגרף i ו j .

- <u>הגדרה</u>:המשקל של המסלול

$$P=(V_0, V_1, V_2, \ldots\ldots\ldots V_k)$$

- הינו סכום המשקולות המיוחסות לקשתות (V_{i-1}, V_i) , לכל $1 < i < k$.
- כלומר

$$W(P) = \sum_{i=1}^{k} W(V_{i-1}, V_i)$$

-
- $W(V_{i-1}, V_i)$ מייצג את המשקל שעל הקשת (V_{i-1}, V_i).
- W(P) מייצג את המשקל של המסלול P .

- נגדיר את אורך (משקל) המסלול הקצר ביותר (המינימלי) מקודקוד u לקודקוד v בגרף על ידי :
- אם קיים מסלול כלשהו P מקודקוד
- u לקודקוד vברשת
-

$$L(u,v)=\begin{cases} \min_{p} W(p) & \\ \infty & \end{cases}$$

- אחרת

- הגדרה:מסלול קצר ביותר מקודקוד **u** אל קודקוד **v** מוגדר כמסלול **p** כלשהו שעבורו $W(p)=L(u,v)$
- עתה נכיר מספר אלגוריתמים שונים למציאת מסלולים קצרים וחד מהם הוא:
- **<u>אלגוריתם דיקסטרה - Dijkstra</u>**
- נתונה רשת $G=(V,E)$ עם פונקציית המשקל $W:E\to R^{+}$
- כלומר, לכל קשת מתאימים משקל <u>חיובי</u>.

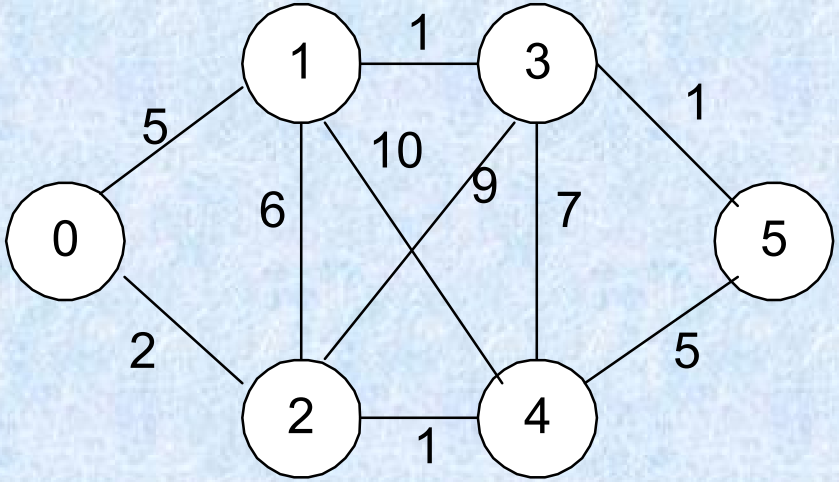
- להלן מספר דרישות לצורך ביצוע האלגוריתם :
- א . נניח שהגרף מיוצג בעזרת מטריצת סמיכות כדלקמן:
- אם קיימת קשת (i , j)
- אם i=j
- אחרת

$$a_{ij} = \begin{cases} E_{ij} \\ 0 \\ \infty \end{cases}$$

- בהנחה שקיימת קשת (i , j) , <u>אורך המסלול המינימלי הזמני</u> מקודקוד i לקודקוד j הינו המספר שמיוחס לקשת (i , j) .
- אורך המסלול המינימלי של המסלול המעגלי מקודקוד i לעצמו הינו 0 , כיוון שלא מאפשרים מעגלים שאורכם שלילי או אפס.
- קביעה זו די הגיונית כי מחפשים מסלולים בעלי אורך מינימלי שהינם מסלולים פשוטים וללא מעגלים.

- במידה ולא קיימת קשת (i , j) , לא ברור שבעתיד יהיה מסלול מקודקוד i לקודקוד j , לכן המרחק המינימלי הקצר ביותר מקודקוד i לקודקוד j הינו המרחק המינימלי הגרוע ביותר שהינו ∞ .
- לדוגמא עבור הרשת הבאה:

מטריצת הסמיכות תוגדר כך:



	0	1	2	3	4	5
0	0	5	2	∞	∞	∞
1	5	0	6	1	10	∞
2	2	6	0	9	1	∞
3	∞	1	9	0	7	1
4	∞	10	1	7	0	5
5	∞	∞	∞	1	5	0

- ב.נניח שקודקוד מקור הינו קודקוד 0.
- ג. קבוצת הקודקודים תחולק לשתי קבוצות:
- אחת הקבוצה P(לכבוד "קבוע") אשר תכיל קודקודים, כך שאורך המסלול המינימלי מקודקוד מקור עד אליהם הינו קבוע ולא ישתנה בעתיד עד סוף האלגוריתם.
- והשניה הקבוצה T (לכבוד "זמניים" temporaries) אשר תכיל קודקודים כך שאורך המסלול המינימלי מקודקוד מקור עד אליהם הינו זמני ועשוי להשתנות בעתיד עד סוף האלגוריתם.

- מאחר שאין מסלולים מעגליים בעלי אורך אי חיובי, המסלול בעל אורך המינימלי מקודקוד מקור לעצמו הינו 0 וערך זה לא ישתנה עד סוף האלגוריתם.
- לכן, בהנחה שקודקוד 0 הינו קודקוד מקור, בתחילת האלגוריתם קודקוד 0 ישתייך לקבוצה P ויתר הקודקודים ישתייכו לקבוצה T .

- ד. עבור כל קודקוד u ברשת נרצה לשמור את אורך המסלול הקצר ביותר מקודקוד מקור 0 לקודקוד u ואותו נסמן על ידי d[u] לכבוד המילה (distance).
- בתחילת האלגוריתם לכל קודקוד u , פרט לקודקוד מקור, נבצע את ההשמה הבאה: $d[u] \leftarrow \infty$ ו- $d[0] \leftarrow 0$ מאחר שאורך המסלול הקצר ביותר מקודקוד מקור לעצמו הינו 0.

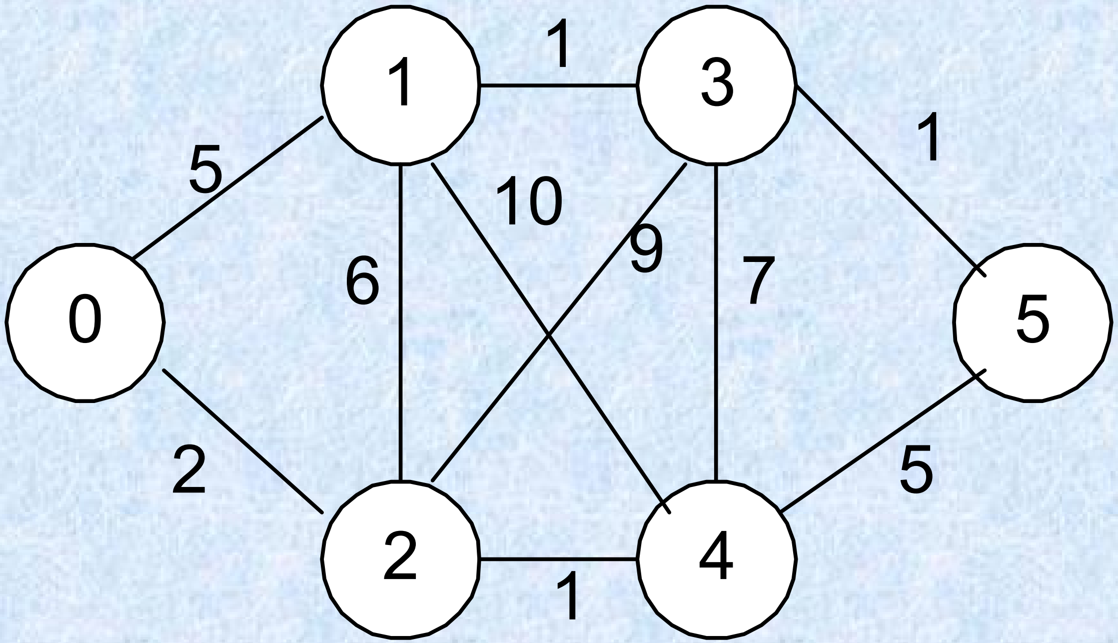
- ה. כמו כן נרצה לשמור לגבי כל קודקוד מידע על זהות הקודקוד הקודם לו ("הורה" שלו) במסלול הקצר.
- לאור זאת נשתמש במערך Pa , כך שלכל קודקוד u ברשת Pa[u] יציין קודקוד ממנו הגענו ל - u .
- <u>הערה</u>- הסימן Pa, נבע מהסיבה ש - Pa[u] מייצג "הורה" (parent) של קודקוד u בעת סריקה למציאת המסלול הקצר.

- בתחילת האלגוריתם לכל קודקוד u , פרט לקודקוד מקור, נבצע השמה ’_’←Pa[u] (משמעותו undefined - עדיין לא מוגדר).לגבי קודקוד מקור 0 נבצע: Pa[0]←nil , המציין שלקודקוד מקור אין אב.
- לאור האמור לעיל בתחילת האלגוריתם ניתן לבצע סדרת ההוראות הבאות:
- - $P \leftarrow \{0\}$ ו $T=\{ 1,2,\ldots\ldots\ldots,n-1 \}$
- - $d[0] \leftarrow 0$

- לכל קודקוד j שאינו קודקוד מקור , כלומר לכל j=1…n-1 בצע:
- $d[j] \leftarrow a_{0j}$ מאחר ש a_{0j} מתאר את אורך המסלול המינימלי **הזמני** העובר דרך הקשת (0,j) .
- $~~P~~[0] \leftarrow nil$ -
- לכל קודקוד שאינו קודקוד מקור, כלומר לכל j=1…n-1 אם קיימת קשת (0, j) אז בצע : Pa[j]←'0'
- אחרת בצע : Pa[j]←'-'

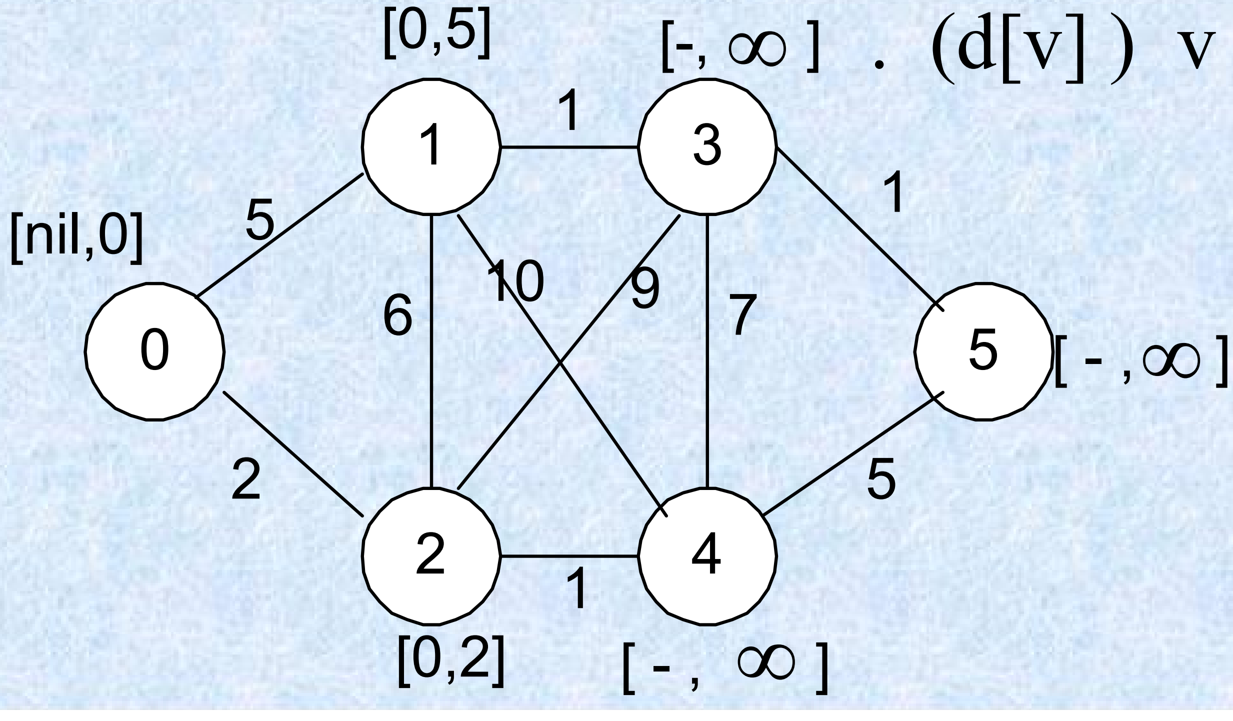
- בשלב הבא עלינו לשפר את אורכי המסלולים הקצרים d[j] מקודקוד מקור 0 לכל קודקוד אחר j , $1 \le j \le n\text{-}1$.
- הדרך לשיפור אורכי המסלולים מתבססת על תהליך איטרטיבי של איתור מסלול מקודקוד מקור לקודקוד j שבעזרתו ניתן לשפר את אורך המסלול המינימלי, עד שלא יהיה מקום לשיפורים נוספים.

טרם נציג את האלגוריתם נדגים את אופן הפעולה של האלגוריתם של דיקסטרה על הרשת הבאה:



- בתהליך התיאור של האלגוריתם סמוך לכל קודקוד V של הגרף מופיעים שני מספרים ;
- השמאלי מייצג את הקודקוד שהינו "הורה" של(Pa[v]) v ;
- הימני מייצג את אורך המסלול הזמני הקצר ביותר מקודקוד מקור 0 לקודקוד v (d[v]) .
- תמונת הרשת

בהתחלה הינה:



- $T=\{ 1,2,3,4,5 \}$ $P=\{ 0 \}$
- עתה השאלה המרכזית היא כיצד משפרים את המסלולים מקודקוד מקור 0 ליתר הקודקודים כך שאורכיהם יהיו מינימליים.

קודקוד v	0	1	2	3	4	5
d[v]	0	5	2	∞	∞	∞
	קבוצה P	קבוצה T				

- בשיטה זו בכל איטרציה נבצע 2 צעדים.
- ***<u>איטרציה ראשונה</u>***
- <u>צעד ראשון</u>
- נמצא קודקוד, מבין הקודקודים "הזמניים" שבקבוצה T , בעל אורך המסלול המינימלי הזמני הקטן ביותר מקודקוד מקור עד אליו. נכנה קודקוד זה בשם K.
- בדוגמא שלנו K=2 , מאחר ש d[2]=2 ול- d[2] ערך הכי קטן מבין כל ה- d[v] עבור $v \in T$.

- כעת נצרף את הקודקוד הזה- K לקבוצה P
- ונוריד אותו מקבוצה T .
- כלומר המרחק המינימלי מקודקוד מקור 0 עד אליו (K) הינו קבוע ולא ישתנה עד סוף האלגוריתם.
- אי לכך בדוגמא שלנו:
- $T=\{\ 1,3,4,5\ \}$ $\quad$ $P=\{\ 0,2\ \}$

- <u>צעד שני</u>
- בצעד זה ננסה לשפר את האורכים של המסלולים הקצרים מקודקוד מקור 0 לכל קודקוד j כאשר $j \in T$ וכל מסלול כזה עובר דרך הקודקוד K=2.
- בדוגמא שלנו:
- <u>בחינת מסלול 0~~~~~1</u>
-

אורך המסלול   אורך הקשת

2   6

$0 \sim\sim\sim\sim\sim 2 \longrightarrow 1$

- אורך המסלול העובר דרך הקודקוד K =2 מקודקוד מקור 0 לקודקוד 1 הינו 8 (= 2+6), לעומת ערך 5 אשר מציין אורך המסלול מקודקוד מקור 0 לקודקוד 1 שאינו עובר דרך הקודקוד K=2.
- מאחר ש 5> 8 אין שיפור באורך המסלול מקודקוד מקור 0 לקודקוד 1, העובר דרך הקודקוד 2.

- <u>בחינת מסלול 0~~~~~>3</u>
- אורך המסלול
-

שהיה עד כה	כעת
∞	11

אורך הקשת   אורך המסלול

$$0 \rightsquigarrow \overset{2}{\downarrow} 2 \xrightarrow{9} 3$$

0~~~~~>2 ——> 3

- לכן אורך המסלול המינימלי 0~~~~~>3 משתנה וערכו 11 ו"ההורה" של קודקוד 3 יהיה קודקוד 2 .

- <u>בחינת מסלול 0~~~~~~>4</u>
- אורך הקשת   אורך המסלול

1   2

0~~~~~~>2 ——> 4

-

שהיה עד כה	כעת
∞	3

לכן אורך המסלול המינימלי 0~~~~~~>4 משתנה וערכו 3

ו"ההורה" של קודקוד 4 יהיה קודקוד 2 .

- <u>בחינת מסלול 0~~~~~>5</u>

אורך המסלול שהיה עד כה	אורך המסלול כעת
∞	∞

אורך הקשת   אורך המסלול

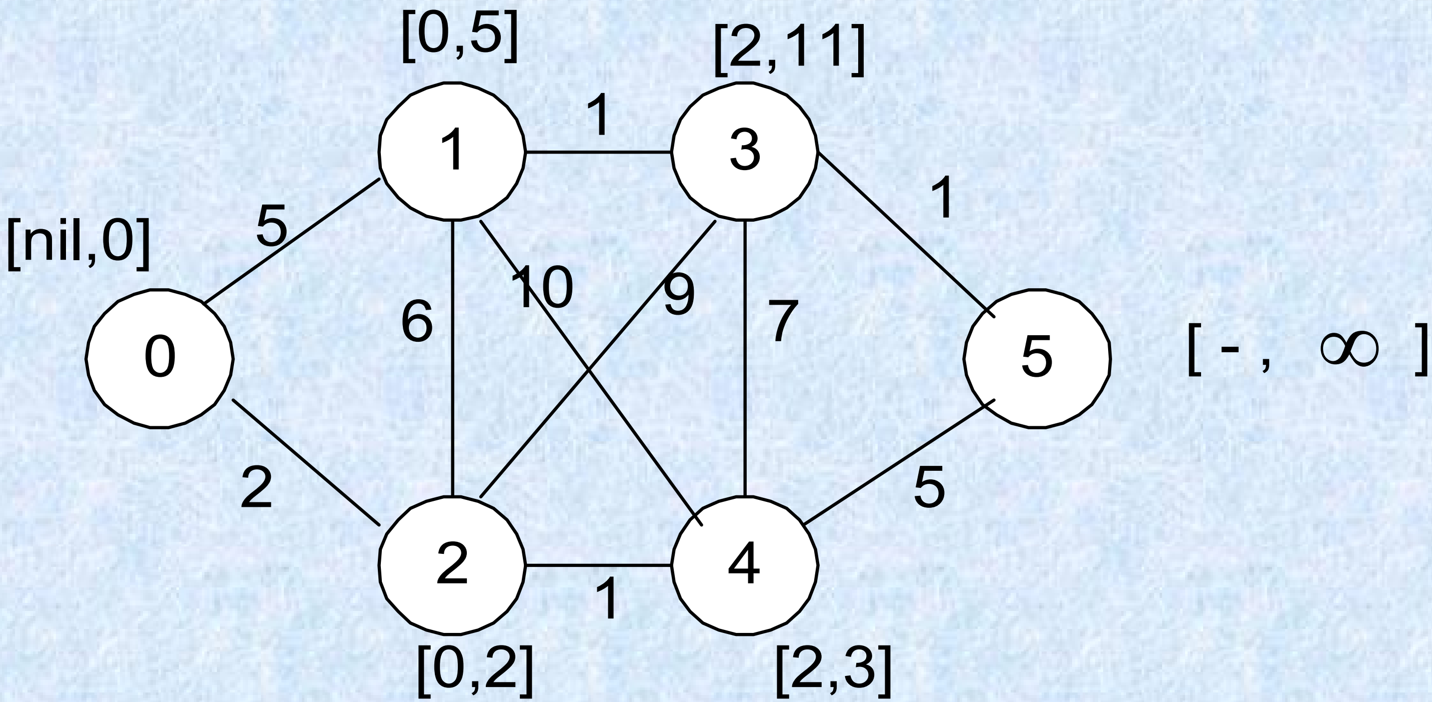
∞ 2

$0 \sim\sim\sim\sim\sim> 2 \xrightarrow{\infty} 5$

- ולכן לא ניתן לשפר את אורך המסלול המינימלי מקודקוד 0 לקודקוד 5.

לאחר בדיקת כל המסלולים האפשריים מקודקוד מקור 0 לכל קודקוד אחר j , כאשר $j \in T$,

תמונת המצב היא:



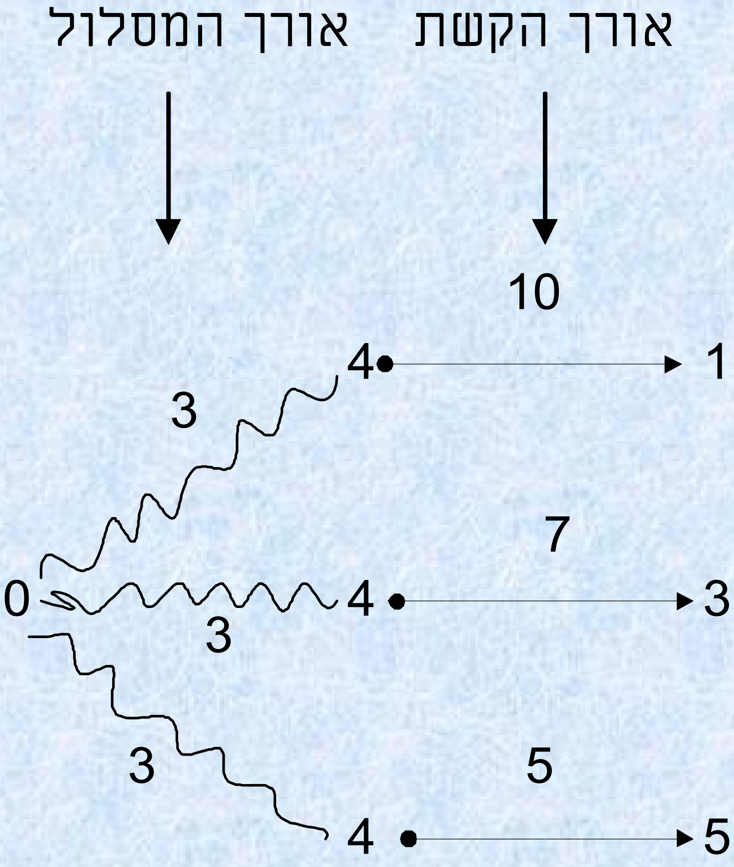
קודקוד v	0	2	1	3	4	5
d[v]	0	2	5	11	3	∞

קבוצה P

קבוצה T

- התהליך חוזר חלילה.

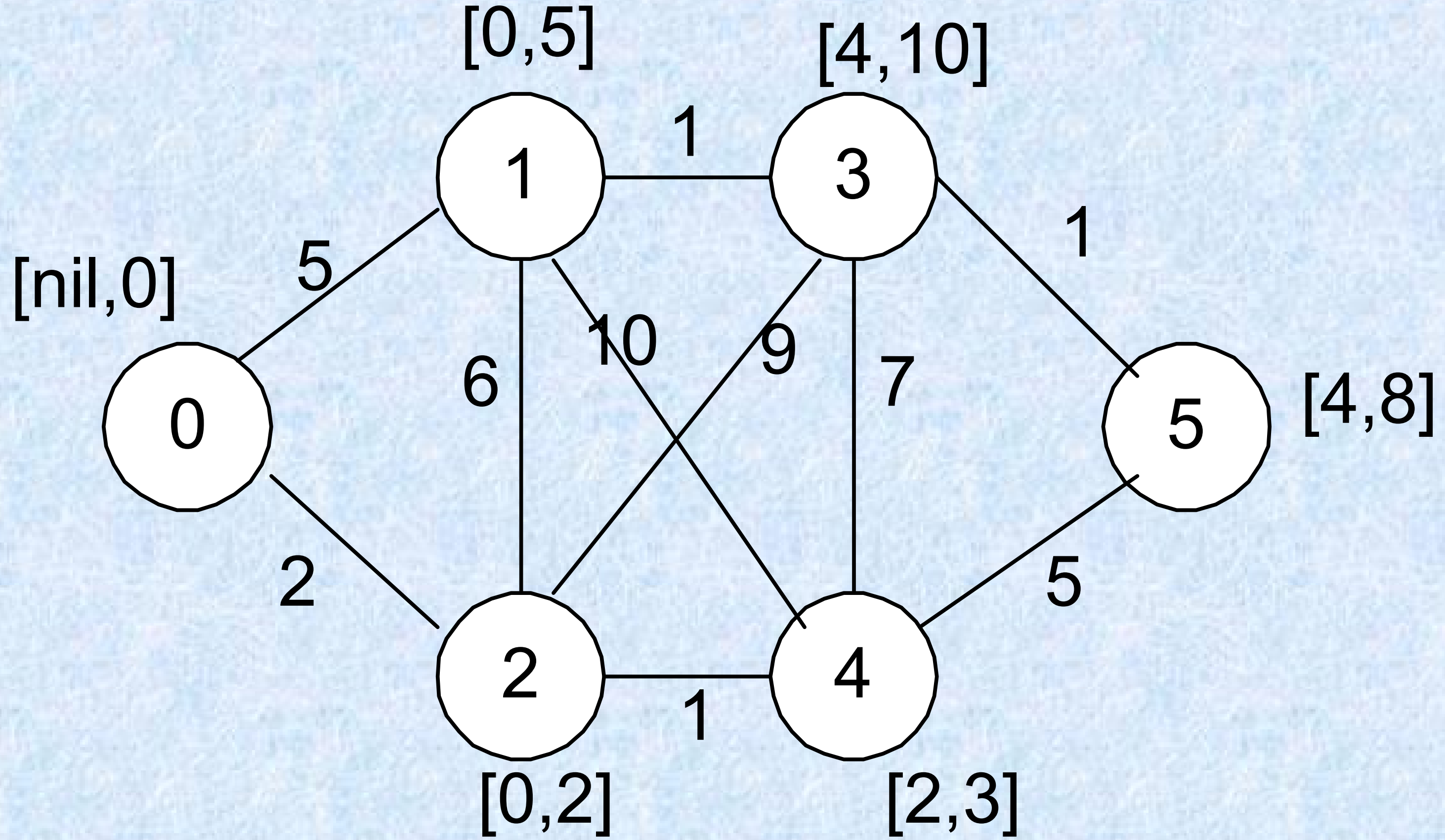
- ***<u>איטרציה שניה</u>***
- <u>צעד ראשון</u>: נקבע $K = 4$, כיוון ש $d[4]=3$ ולמשתנה זה ערך הכי קטן מאשר לכל משתנה אחר $d[j]$ לכל $j \in T$.
- לכן נקבל: $T=\{ 1,3,5 \}$ $P=\{ 0,2,4 \}$.
- <u>צעד שני</u>
- שיפורי מסלולים קצרים מקודקוד מקור 0 לכל קודקוד j , $j \in T$, ומסלולים אלה עוברים דרך הקודקוד $K = 4$.



אורך המסלול שיהיה	אורך המסלול שהיה	אורך המסלול שכעת
5	5	13
10	11	10
8	∞	8

- בחינת מסלול 0~~~~~~~~>1 : רואים שאין שיפור
- בחינת מסלול 0~~~~~~~~~~>3: יש שיפור ולכן ההורה של הקודקוד 3 יהיה קודקוד 4.
- בחינת מסלול 0~~~~~~~~~>5: יש שיפור ולכן ההורה של הקודקוד 5 יהיה קודקוד 4 .

עתה תמונת המצב הינה:

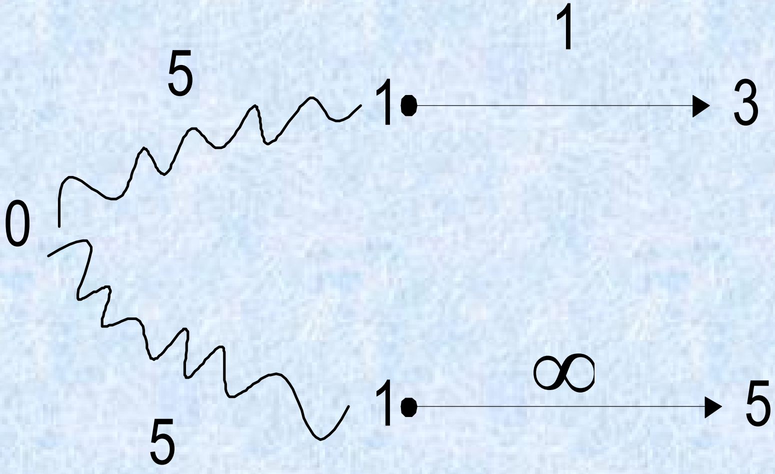


קודקוד v	0	2	4	1	3	5
d[v]	0	2	3	5	10	8
		קבוצה P			קבוצה T	

- התהליך חוזר חלילה.

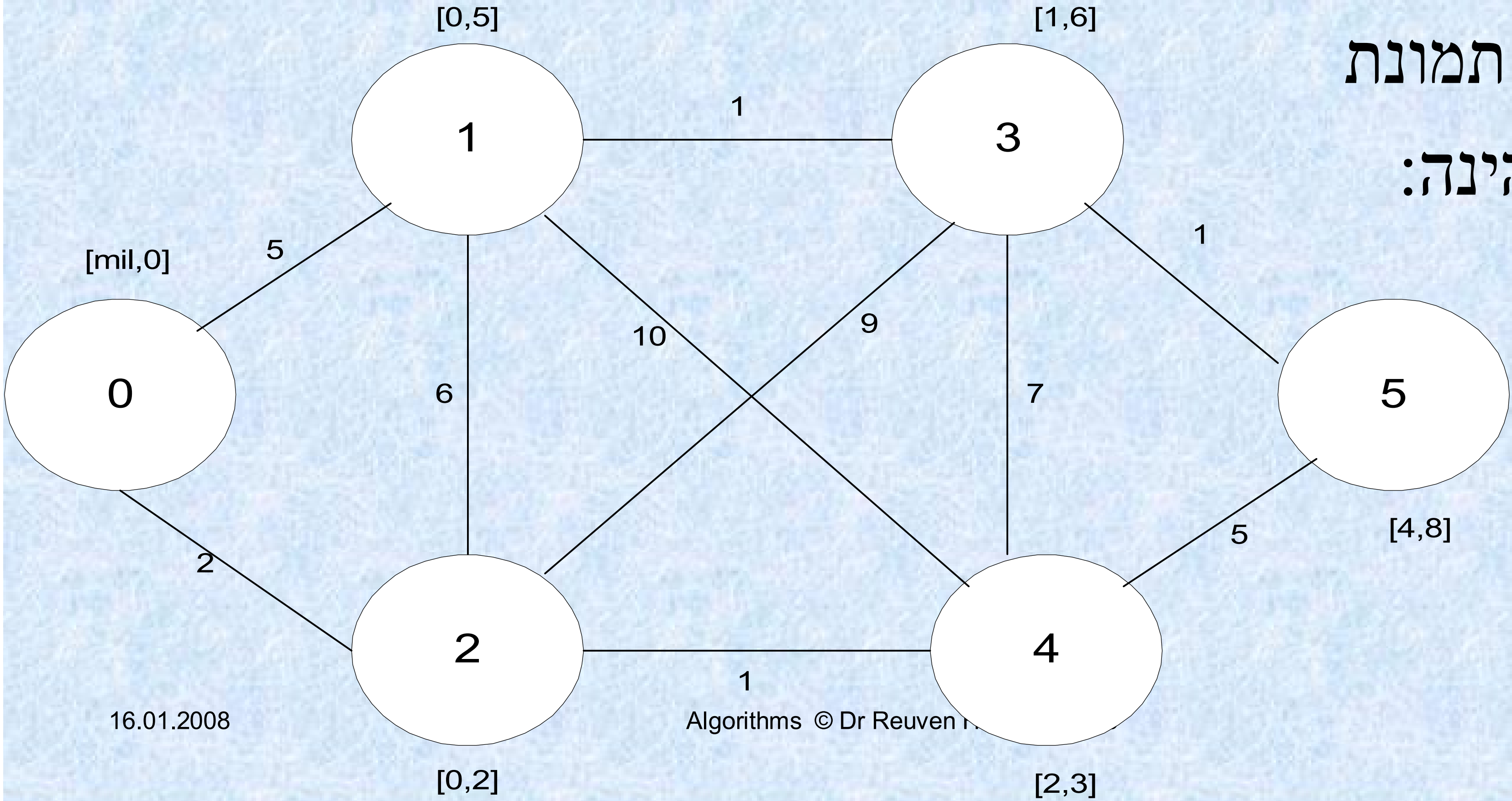
- ***איטרציה שלישית***
- <u>צעד ראשון</u> נקבע $K = 1$, כיוון ש $d[1] = 5$ ולמשתנה זה ערך הכי קטן מאשר לכל משתנה אחר d[j] לכל $j \in T$.
- לכן נקבל $P = \{0,1,2,4\}$ $T = \{3,5\}$
- <u>צעד שני</u> שיפורי מסלולים קצרים מקודקוד 0 לכל קודקוד j, $j \in T$, דרך הקודקוד $K = 1$.

אורך המסלול אורך הקשת



אורך המסלול שיהיה	אורך המסלול שהיה	אורך המסלול שכעת
6	10	6
8	8	∞

- בחינת מסלול 0~~~>3:יש שיפור. ההורה של 3 יהיה 1.
- בחינת מסלול 0~~~~~~~>5: רואים שאין שיפור.
- עתה תמונת המצב הינה:



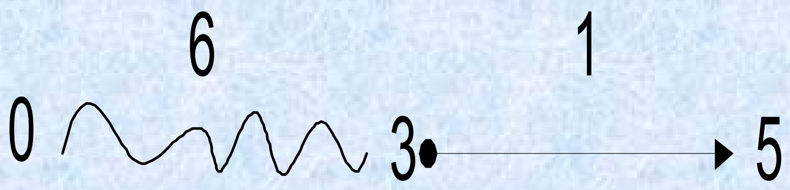
V קודקוד	0	1	2	4	3	5
d[V]	0	5	2	3	6	8

קבוצה P

קבוצה T

- התהליך חוזר חלילה.

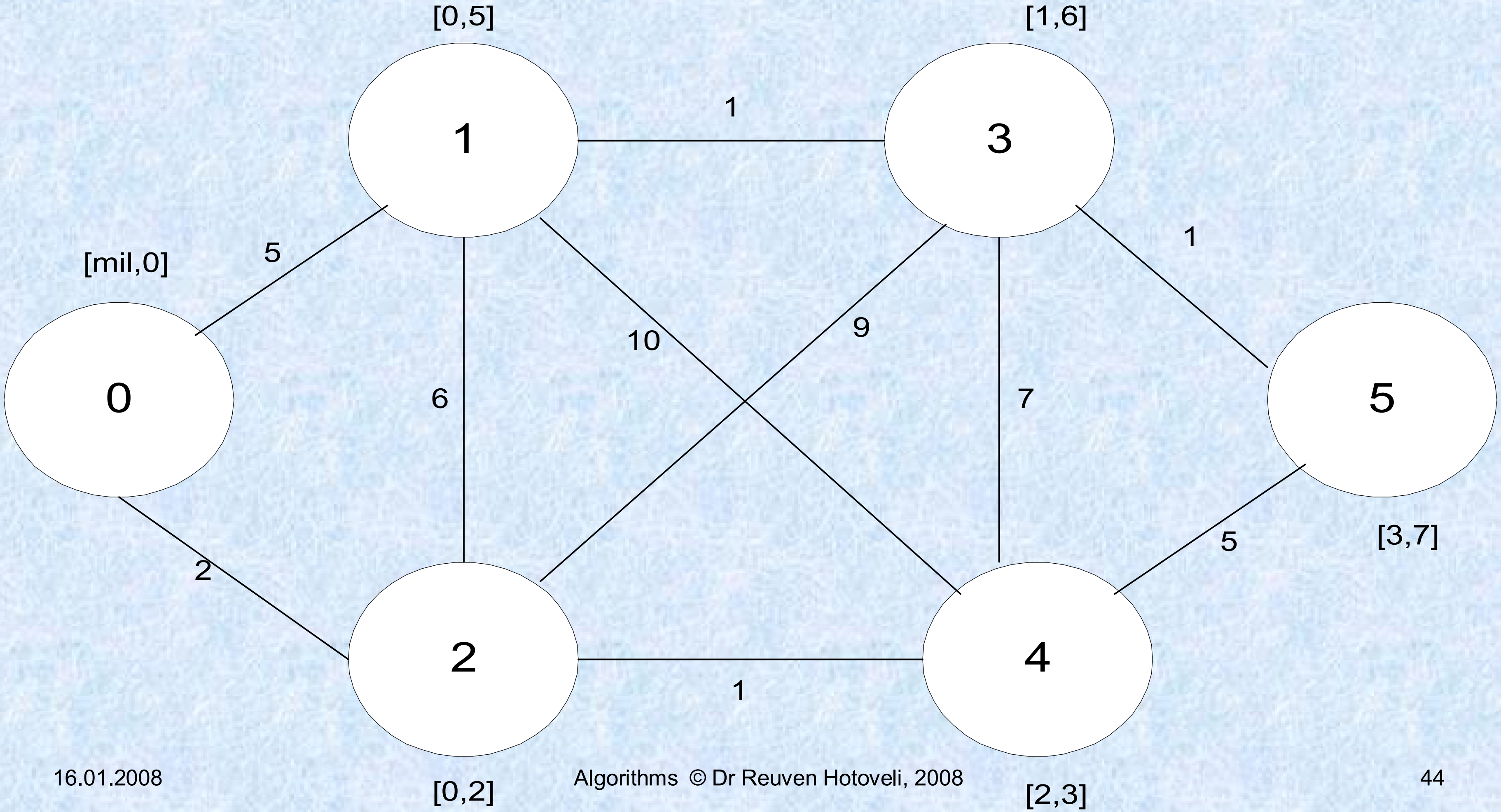
- <u>איטרציה רביעית</u>
- <u>צעד ראשון</u> נקבע $K = 3$, כיוון ש- $d[3] = 6$ ולמשתנה זה ערך יותר קטן מאשר למשתנה $d[6]$ שערכו 8.
- לכן נקבל: $P=\{0, 1, 2, 3, 4\}$ $T=\{5\}$
- <u>צעד שני</u> ניסיון לשיפור מסלול 5 ~~~~> 0 העובר דרך קודקוד $K = 3$.



אורך המסלול שיהיה	אורך המסלול שהיה	אורך המסלול שכעת
7	8	7

- מאחר שיש שיפור באורך המסלול מ- 0 לקודקוד 5 דרך הקודקוד 3 , אז ההורה של קודקוד 5 יהיה קודקוד 3.

- עתה תמונת המצב הינה:

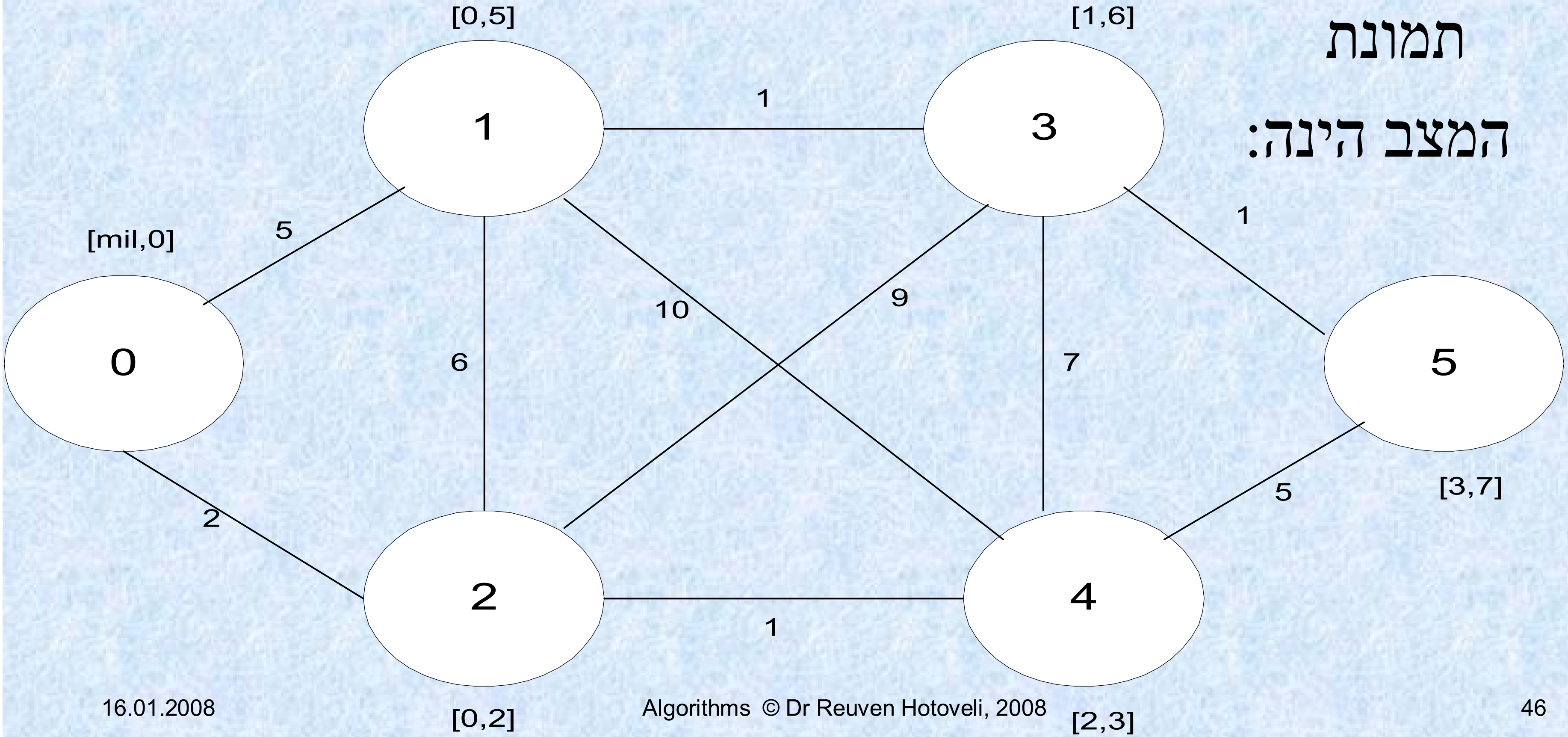


V קודקוד	0 1 2 3 4	5
d[V]	0 5 2 6 3	7
	קבוצה P	קבוצה T

- איטרציה חמישית
- צעד ראשון נקבע ש $K = 6$ (ברור!)
- לכן נקבל: (קבוצה ריקה) - $T=\emptyset$ $\quad p=\{ 1,2,3,4,5, 6 \}$

◈ מאחר ו T היא קבוצה ריקה אזי כל המסלולים הקצרים נקבעו ואין את מה לשפר, לכן האלגוריתם הסתיים וסופית תמונת

המצב הינה:



- הערה: באמצעות האלגוריתם ניתן לקבוע מהו מסלול עצמו.
- כך למשל עבור המסלול 5 ~~~~~~~~> 0 המסלול הינו (מהסוף להתחלה) קודם קודקוד 5 , אביו של 5 הינו 3 ואביו של 3 הינו 1 ואביו של 1 הינו קודקוד 0 ולקודקוד 0 אין אב כיוון שהוא קודקוד מקור.
- לכן המסלול הינו: $0 \longrightarrow 1 \longrightarrow 3 \longrightarrow 5$.

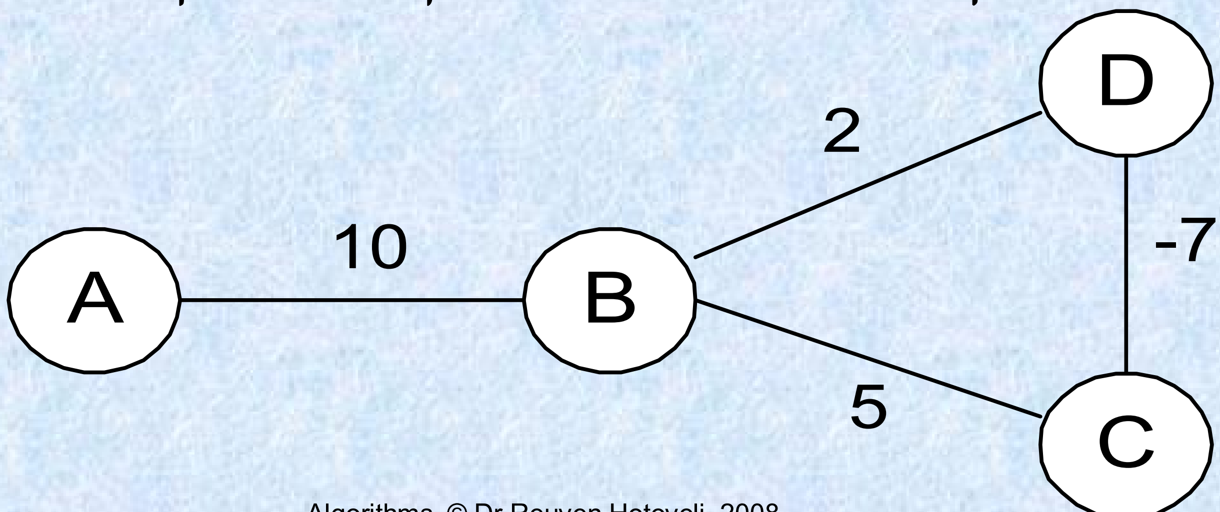
- עתה נוכל לסכם את האלגוריתם כדלהלן:
- <u>צעד 0</u>
- 0.1 $T=\{1,2,\ldots.,n-1\}$ $P=\{0\}$
- 0.2 $d[0] \leftarrow 0$
- 0.3 לכל קודקוד , שאינו קודקוד מקור , כלומר לכל
  $j=1,\ldots..,n-1$ בצע: $d[j] \leftarrow a[0,j]$
- 0.4 סוף הלולאה

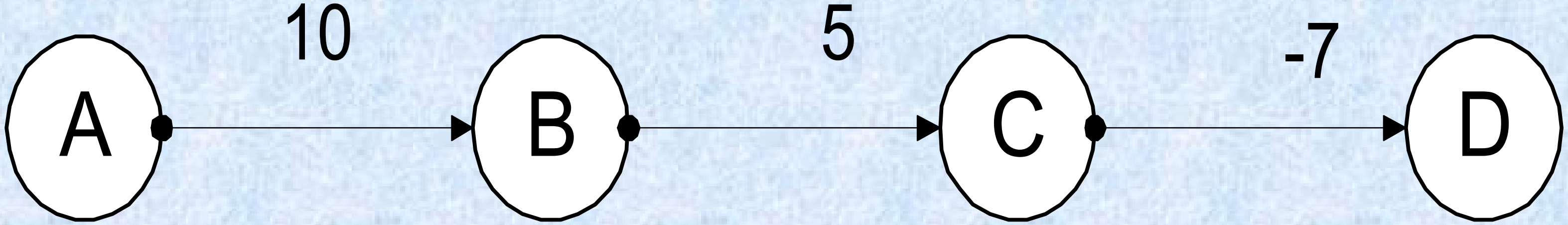
- 0.5 Pa[0]←nil
- 0.6 לכל קודקוד j=1,….,n-1 בצע:
- אם קיימת קשת (0,j)
- אז בצע: Pa [j]←’0’
- אחרת בצע: Pa[j]←’-‘
- 0.7 סוף לולאה.

- <u>צעד 1</u>
- 1.1 מצא קודקוד k מתוך קבוצת הקדקודים "הזמניים"- T
  בעל ערך d[k] מינימלי, כלומר: לכל $j \in T$
  d[k]=min { d[j] }
- 1.2 צרף את הקודקוד k לקבוצה p כלומר p←p+{k}
- 1.3 להוריד את הקודקוד k מקבוצה T כלומר T←T-{k}
- 1.4 אם T=∅ (T הינה קבוצה ריקה) אזי סיים !
- אחרת עבור לצעד 2.

- <u>צעד 2</u>
- 2.1 לכל קודקוד $j \in T$ בצע :
- 2.1.1 אם $d[k]+a[k]]j] < d[j]$
- אז בצע: $Pa[j] \leftarrow k$
- 2.1.2 $d[j]=\min \{ d[j] , d[k]+ a[k][j] \}$
- 2.2 סוף הלולאה
- 2.3 חזור לצעד 1 .

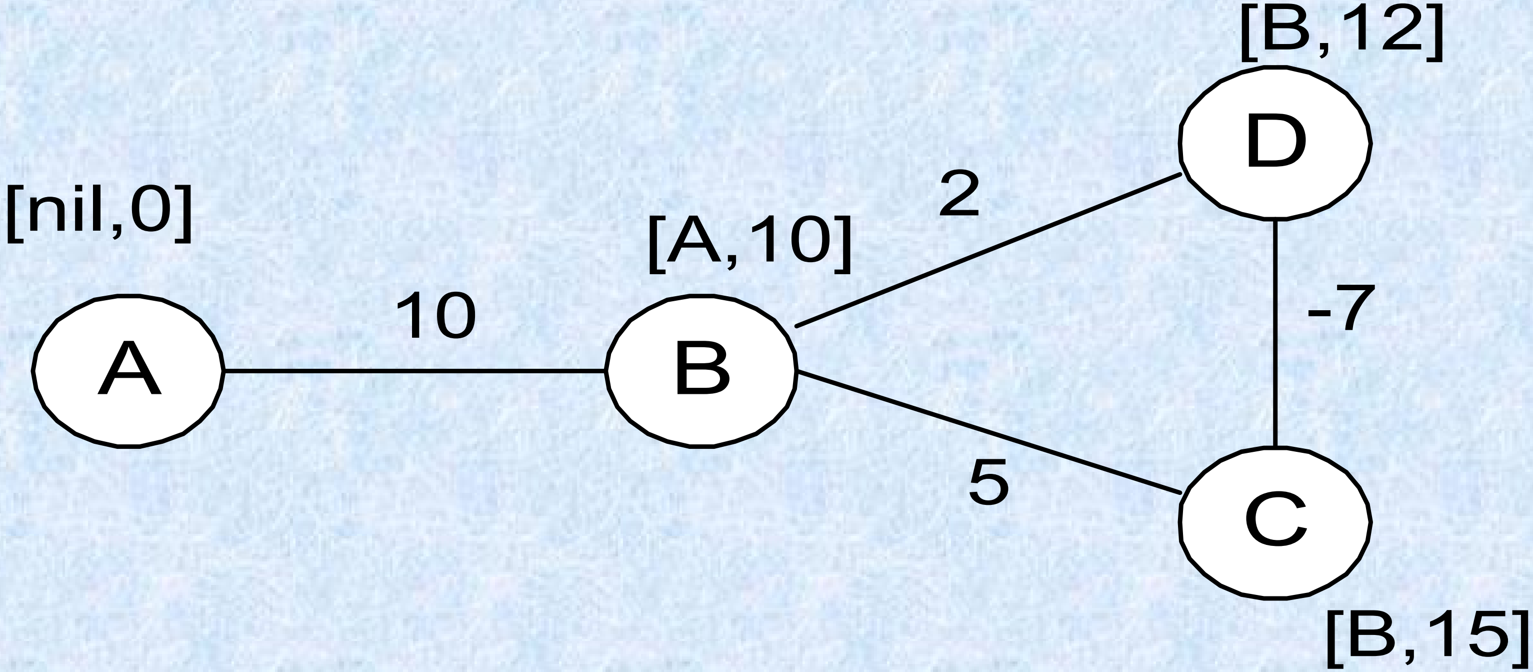
- הערה חשובה! אלגוריתם של דיקסטרה פועל כהלכה בתנאי שכל המשקלות המיוחסות לקשתות הגרף הינן חיוביות.
- נראה זאת בשלילה. **נניח שאפשר לייחס משקל שלילי לקשת כלשהי**. נתבונן על הגרף הבא:



- שים לב שלקשת (C,D) מיוחס מספר שלילי (7 -). קל לראות שהמסלול בעל אורך הקצר ביותר מקודקוד A לקודקוד D הינו:
- 

- ואורכו 8.

- אך בעזרת אלגוריתם דיקסטרה לאחר שתי איטרציות ראשונות נקבל את תמונת המצב הבא (בדוק!):



- באיטרציה הבאה (מבין הקודקודים הזמניים C , D) נבחר בקודקוד D, כיוון ש $d[D]= 12 < d[C]=15$.
- לכן הקבוצה P תהיה : $P=\{A, B,D \}$
- כלומר אורך המסלול המינימלי מ- A ל- D הוא 12 והוא "לכאורה" קבוע, וערכו לא ישתנה עד סוף האלגוריתם, כי הקודקוד D מצטרף לקבוצה P.
- ברור שהתוצאה שקיבלנו $d[D] = 12$ ***<u>אינה נכונה</u>***, כיוון שהערך הצפוי ל - $d[D]$ הינו 8 .זוהי סתירה להנחתנו.

- מסקנה: אם ברצוננו להפעיל את האלגוריתם של דיקסטרה חובה לדרוש שכל המשקולות על הקשתות חייבות להיות חיוביות.
- <u>יעילות האלגוריתם של דיקסטרה</u>
- נתון גרף $G = (V,E)$. $|V|$ מציין את מספר הקדקודים בגרף G.
- <u>צעד 0</u> סיבוכיות זמן הריצה של הצעד 0 הינה $O(|V|)$.

צעד 1

- 1.1 נניח שהקבוצה T ממומשת בעזרת מערך.
- לאור הנחה זו צעד זה דורש זמן $0(|V|)$.
- 1.2 נניח שהקבוצה P ממומשת בעזרת מערך. לכן צעד זה דורש זמן $0(1)$.
- 1.3 בהמשך להנחה שב – 1.1 בצעד 1.1 ניתן לשמור מידע על מיקומו של הקודקוד K . לכן, צעד זה דורש $0(1)$.
- מכיוון שצעד 1 מתבצע $|V|$ פעמים,אז הזמן הכולל שצעד 1 דורש הוא $0(|V|^2)$.

- <u>צעד 2</u>
- נניח שהגרף מיוצג בעזרת רשימות סמיכות .
- ברור כי באלגוריתם הנדון , כל קודקוד של גרף מוכנס לקבוצה P פעם אחת כך שכל קשת ברשימת הסמיכות נבחנת בדיוק פעם אחת במהלך האלגוריתם.
- לאור האמור לעיל צעד 2 מתבצע בסך הכל $0(|E|)$ פעמים.

- סופית:סיבוכיות זמן הריצה של האלגוריתם "דיקסטרה" היא: $\boxed{0(|V|^2 + |E|) = 0(|V|^2)}$
- **<u>הערה:</u>**
- למעשה ניתן להשיג זמן ריצה של $0(|E|\log|V|)$ כאשר הקבוצה T ממומשת בעזרת מבנה נתונים מסויים הנקרא ערמה בינרית (heap) .
- להלן מספר עובדות אודות הערמה :

- לבנות ערמה דורשת $0(|V|)$.
- איתור וסילוק האיבר הקטן ביותר שבערמה דורשת
- זמן $0(\log|V|)$.
- צעד ~~2.1~~ מתבצע בזמן $0(\log|V|)$.
- <u>צעד 1</u>
- מתבצע $|V|$ פעמים ובכל צעד נדרש זמן $0(\log|V|)$.

  לכן הזמן הכולל של צעד 1 הוא $0(|V|\log|V|)$.

- <u>צעד 2</u>
- עדיין כל קודקוד מוכנס לקבוצה P בדיוק פעם אחת .
- לכן כל קשת ברשימת הסמיכות נבחנת בדיוק פעם אחת במהלך האלגוריתם .
- כאמור מס' הקשתות הכולל ברשימת הסמיכות הוא $0(|E|)$.
- לכן צעד 2 מתבצע $0(|E|)$ פעמים ובכל צעד נדרש זמן $0(\log|V|)$ עדכון הערך והכנסתו לערמה מחדש עם הערך החדש. לכן זמן הריצה של צעד 2 הוא $0(|E|\log|V|)$.

- סופית , זמן הריצה של האלגוריתם כולו הוא :

$$0(|E|\log|V| + |V|\log|V|) = 0((|E| + |V|)\log|V|)$$

- בגרף קשיר, מאחר ש $|E| \geq |V|$, סיבוכיות זמן הריצה של האלגוריתם דיקסטרה היא: $0(|E|\log|V|)$.
- רואים כי כדאי לממש את T כערמה בעבור גרפים דלילים בלבד.
- הערה חשובה: קיים מבנה נתונים מתקדם הנקרה ערמת פיבונצ'י ובאמצעותו ניתן להשיג זמן ריצה O(VLogV+E)
- וזהו מימוש מהיר יותר אסימפטוטית.

ניסוח אחר של האלגוריתם

- INITIALIZE-SINGLE-SOURCE(G,s)
- 1. for each vertex v∈V do
-     1.1 d[v]← ∞
-     1.2 $\pi[v] \leftarrow NIL$
- 2. d[s]← 0
- כאשר $\pi[v]$ הוא קודקוד "קודם" של v .

המשך ניסוח חדש

- טכניקת ההקלה (relaxation) :

- RELAX(u,v,w)
- 1. if d[v] > d[u] + w(u,v) then
- 1.1 d[v] ← d[u] + w(u,v)
- 1.2 $\pi[v] \leftarrow u$

המשך הניסוח

- DIJKSTRA(G,w,s)
- 1. INITIALIZE-SINGLE-SOURCE(G,s)
- 1. S← $\emptyset$
- 2. Q← V {Q - הוא תור של קודקודים}
- 3. While Q $\neq \emptyset$ do
- 3.1 u← Extract_Min(Q)
- 3.2 $S \leftarrow S \cup \{u\}$
- 3.3 for each vertex $v \in Adj[u]$
- do RELAX(u,v,w)

נכונות האלגוריתם

- משפט 1:
- נתון גרף $G=(V,E)$ עם פונקצית המשקל $W:E\rightarrow R$.
- יהי $P=(V_0,V_1,\ldots,V_k)$ מסלול הקצר ביותר מקודקוד V_0 לקודקוד V_k.
- אם $Pij=(V_i,V_{i+1},\ldots,V_j)$ הינו תת מסלול מקודקוד V_i לקודקוד V_j עבור $0\leq i\leq j\leq k$, אז Pij הינו מסלול בעל אורך מינימלי מקודקוד V_i לקודקוד V_j.

- הוכחה:
- נפרק את המסלול P לתתי מסלולים הבאים:
- P_{0i} P_{ij} P_{jk}
- $V_0 \sim V_i \sim V_j \sim V_k$
- ברור כי
- $W(P)=W(P_{0i})+W(P_{ij}) + W(P_{jk})$

- כעת נניח שקיים מסלול אחר P'_{ij} מקודקוד V_i לקודקוד V_j המקיים: $W(P'_{ij}) < W(P_{ij})$
- עתה נתבונן במסלול החדש מקודקוד V_0 לקודקוד V_k
- P_{0i} P'_{ij} P_{jk}
- $V_0 \sim\sim V_i \sim\sim V_j \sim\sim V_k$
- המשקל של המסלול הינו :
- $W(P') = W(P_{0i}) + W(P'_{ij}) + W(P_{jk}) < W(P_{0i}) + W(P_{ij}) + W(P_{jk})$

- כלומר $W(P') < W(P)$.
- וזוהי סתירה לנתון ש P – מסלול בעל אורך מינימלי.
- לכן הנחתנו אינה נכונה .
-
- מש"ל.

- מסקנה 1 :
- נתון גרף G=(V,E) עם פונקצית משקל $W : E \rightarrow R^{+}$
- נניח כי s הינו קודקוד מקור.
- יהי P מסלול בעל אורך מינימלי מקודקוד s לקודקוד a בגרף.
- מסלול זה P ניתן לפירוק באופן הבא:
- $$s \overset{p'}{\rightsquigarrow} v \longrightarrow a$$
- ומתקיים : L(s,a)=L(s,v)+W(v,a)

- כלומר, אורך המסלול המינימלי מקודקוד s לקודקוד a הינו אורך המסלול המינימלי מקודקוד s לקודקוד v בתוספת המשקל שעל הקשת (v,a) , בתנאי שקיימת קשת מקודקוד v לקודקוד a .
- הוכחה:
- מאחר ש - P הוא מסלול בעל אורך מינימלי אז לפי ההגדרה $L(s,a)=W(P)$

- אך ברור כי :
- $W(P)=W(P')+W(v,a)$
- כמו כן לפי משפט 1ברור כי P' הינו מסלול בעל אורך מינימלי מקודקוד s לקודקוד v .
- כלומר מתקיים: $W(P')=L(s,v)$
- לסיכום
- $L(s,a)=W(P)=W(P')+W(v,a)=L(s,v)+W(v,a)$

- מסקנה 2:
- לכל קשת $(u,a) \in E$ ברשת מתקיים :
- $L(s,a) \leq L(s,u)+W(u,a)$

- מסקנה 3:
- תכונה הנובעת מאלגוריתם דיקסטרה
- תהי $(u,v) \in E$ קשת ברשת. בתום צעד מספר 2 של האלגוריתם דיקסטרה מתקיים: $d[v] \leq d[u] + w(u,v)$

- הוכחה:
- בצעד מספר 2 של האלגוריתם ביצענו את המשפט הבא :
- $d[v] = \min \{ d[v], d[u]+w(u,v) \}$
- אם $d[v] > d[u]+w(u,v)$ אזי לאחר צעד 2 יתקיים:
- $d[v] = d[u]+w(u,v)$
- ואם $d[v] <= d[u] +w(u,v)$ אזי $d[u]$ ו-$d[v]$ אינם משתנים ולכן $d[v] <= d[u]+w(u,v)$.
  מש"ל .

- <u>משפט 2</u>
- נתון גרף G=(V,E) עם פונקצית משקל $W : E \to R^+$.
- יהי s קודקוד מקור באלגוריתם למציאת מסלולים בעלי משקל מינימלי .
- בתחילת האלגוריתם מתקיים: $d[v] \geq L(s,v)$ לכל קודקוד v בגרף .
- אם $d[v] = L(s,v)$ אזי הערך של d[v] לא ישתנה עד סוף האלגוריתם.

- הוכחה
- לפי צעד 0 באלגוריתם $d(s)=0$.
- מאחר שאנו לא מרשים קיומם של המעגלים באורך שלילי ברשת אז אורך המסלול הקצר ביותר מקודקוד מקור לעצמו הינו 0 , כלומר $L(s,s)=0$ ולכן $d[s]=L(s,s)$.
- בעבור יתר הקודקודים, פרט למקור, $d[v]=\infty$ ולכן מתקיים $d[v] \geq L(s,v)$.

- עתה נניח שקודקוד v הוא הקודקוד הראשון שעבורו מתקיים: $d[v] < L(s,v)$
- בתום צעד 2 באלגוריתם של דיקסטרה העדכון של $d[v]$ התבצע באמצעות הקשת (u,v) .
- נתבונן במסלול הבא: $s \rightsquigarrow u \longrightarrow v$
- $d[u]+w(u,v)=d[v] < L(s,v) \le L(s,u)+w(u,v)$
- $\uparrow$ הנחה $\uparrow$ לפי מסקנה 2

- מכאן נובע ש $d[u] < L(s,u)$
- זוהי סתירה להנחה ,כיוון שכאשר בוחנים את הקשת (u,v) משנים את $d[v]$ ולא את $d[u]$.
- לפני קביעת ערכו ל $d[v]$, נקבע ערכו של $d[u]$
- מאחר שמניחים שקודקוד v הוא הקודקוד הראשון שעבורו מתקיים $d(v) < L(s,v)$ לכן לא יתכן כי $d(u) < L(s,u)$.

- מאחר שקיבלנו סתירה להנחתנו, המסקנה היא ש-
- $d[v] \geq L(s,v)$ לכל קודקוד v ברשת .
- לכן אם באיטרציה מסוימת משיגים את השיווין $d[v]=L(s,v)$
- ***הוא אינו יכול לקטון***, משום שזה עתה ראינו כי $d[v] \geq L(s,v)$
- ***והוא אינו יכול לגדול*** כי צעד 2 של האלגוריתם
- $d[v]=\min \{ d[v] , d[u]+w(u,v) \}$
- אינו מגדיל את ערכו של d. ***מש"ל***

משפט 3

יהי מסלול בעל אורך מינימלי $s \rightsquigarrow u \longrightarrow v$

בגרף $G=(V,E)$. אם $d[u]=L(s,u)$ אז $d[v]=L(s,v)$.

הוכחה

$$d[v] \underset{\text{מסקנה 3}}{\leq} d[u]+w(u,v) \underset{\text{נתון}}{=} L(s,u)+w(u,v) \underset{\text{לפי מסקנה 1}}{=} L(s,v)$$

אך לפי משפט 2 $d[v] \geq L(s,v)$ לכן $d[v] = L(s,v)$.

מש"ל.

- <u>משפט 4</u>
- (משפט מרכזי להוכחת נכונות האלגוריתם של דיקסטרה).
- לאחר הרצת אלגוריתם של דיקסטרה על הגרף G=(V,E), עם פונקצית משקל W (המשקולות על קשתות חיוביות) וקודקוד s שהינו קודקוד מקור, לכל קודקוד $u \in V$ מתקיים: d[u]=L(s,u)

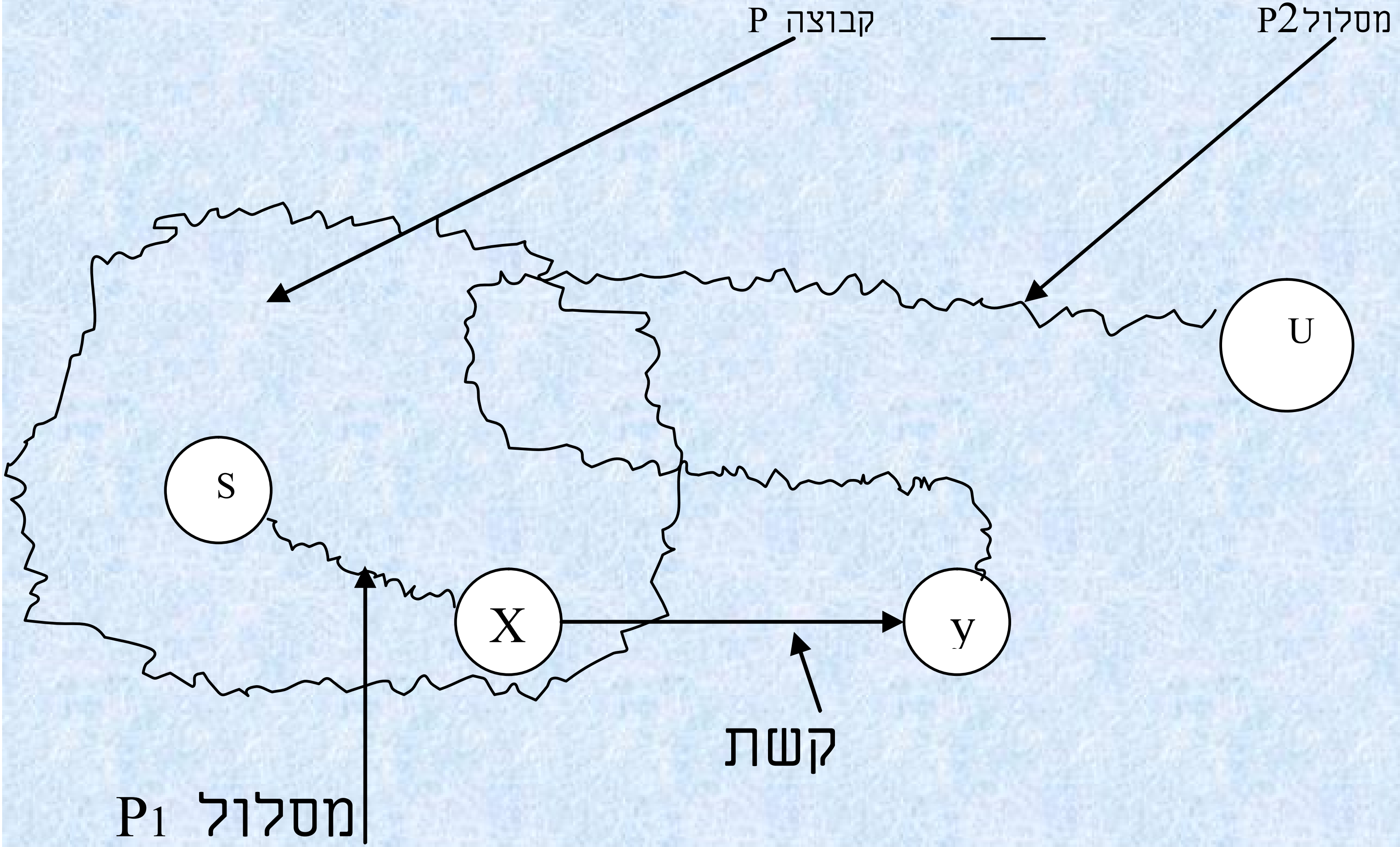
הוכחה

- נראה,עבור כל קודקוד u , שמתקיים $d[u]=L(s,u)$ כאשר הקודקוד u מצטרף לקבוצה P (צעד מספר 1 באלגוריתם).
- כאמור P זוהי קבוצת הקדקודים כך שהמסלול בעל אורך מינימלי מקודקוד מקור S עד אליהם הינו <u>קבוע</u> ולא ישתנה בעתיד עד סוף האלגוריתם.

- נניח בשלילה ויהי u ***הקודקוד הראשון*** שעבורו $d[u] \neq L(s,u)$ לאחר שהקודקוד u הצטרף לקבוצה P.
- ברור כי $u \neq s$ מכיוון ש- s הינו קודקוד מקור והוא הראשון שמצטרף לקבוצה P ולפי צעד 0 של האלגוריתם מתקיים: $d[s] = 0 = L(s,s)$

- מאחר ש- $u \neq s$ ו- $s \in P$ אזי ברור שהקבוצה P שונה מקבוצה ריקה ($P \neq \emptyset$) לפני שהקודקוד u מצטרף לקבוצה P .

- יהי מסלול מקודקוד $s \in P$ לקודקוד $u \in T$ (סימונים לפי האלגוריתם) המתואר באיור הבא :


קבוצה P
מסלולP2
U
S
X
y
קשת
מסלול P1

- נניח שהמסלול המינימלי מקודקוד s לקודקוד u הינו
- P1 P2
- $s \rightsquigarrow x \longrightarrow y \rightsquigarrow u$
- כאשר $x \in P$ ו $y \in T$.
- מאחר ש- $x \in P$ וקודקוד u <u>***הוא הראשון***</u> שעבורו
- $d[u] \neq L(s,u)$, כאשר u מצטרף לקבוצה P אזי ברור כי $d[x]=L(s,x)$

- בזמן שהקודקוד x הצטרף לקבוצה P (בצעד מס' 1 של האלגוריתם) מייד לאחר מכן צעד מס' 2 של האלגוריתם מעדכן את ה- $d[y]$ של הקודקוד y ולפי משפט 3 נובע כי $d[y]=L(s,y)$.
- המשמעות היא ברגע ש- u יצטרף לקבוצה P ברור שצריך להתקיים : $d[y]= L(s,y)$.

- מאחר שהקודקוד y הינו קודקוד מקדים לקודקוד u במסלול הקצר מקודקוד s לקודקוד u ולפי דרישת האלגוריתם **כל המשקולות הן חיוביות** אזי ברור כי :
- $L(s,y) \leq L(s,u)$ ולכן מתקיים :
- $d[y]=L(s,y) \leq L(s,u) \leq d[u]$
-

לפי משפט 2

- הקדקודים y ו- u שייכים לקבוצה T, לכן כאשר קודקוד u נבחר ברור שצריך להתקיים:
- $d[u] \le d[y]$
- מחד $d[u] \ge d[y]$ ומאידך $d[u] \le d[y]$.
לכן,ברור שצריך להתקיים $d[u] = d[y]$.
- $d[y] = L(s,y) \le L(s,u) \le d[u] = d[y]$
-

ראינו עתה

- ולכן
- $d[y]= L(s,y)=L(s,u)=d[u]=d[y]$
- כלומר $d[u]=L(s,u)$ וזוהי סתירה להנחתנו .
- ולכן כל קודקוד u שמצטרף לקבוצה P מתקיים
- $d[u]=L(s,u)$
- ועל פי משפט 2 ידוע שה- $d[u]$ לא ישתנה עד סוף האלגוריתם . ***<u>מש"ל</u>***

תרגול כיתה

- נתון גרף מכוון $G=(V,E)$, $w: E \rightarrow \{0,1,2\}$, $s \in V$.

מצא מסלולים קצרים ביותר מ-s.

הרעיון

- נריץ את האלגוריתם של Dijkstra וננצל את העובדה שהמשקולות הם רק 0, 1, ו-2 כדי לממש את תור העדיפויות בצורה יעילה יותר.
- תזכורת: זמן הריצה של Dijkstra הוא $O(|V|+|V|*extract\_min + |E|*update)$.

אורך המסלולים הוא O(|V|)

- כיוון שהמשקולות אי-שליליים, המסלולים הקצרים ביותר חייבים להיות פשוטים.
- מכיוון שמסלול פשוט מכיל לכל היותר 1- | V | קשתות, ומשקל מקסימלי של קשת הוא 3, הערך d של כל צומת הוא מספר שלם בין 0 ל-|V|3 (או אינסוף).

- לשם פשטות, נחליף את הסימון $d = \infty$ בסימון $d=3|V|+1$, ונחליפו בחזרה כאשר האלגוריתם יסתיים.
- כלומר, עלינו לממש ADT בעל ממשק דומה לזה של ערימה (heap), כאשר נתון שהמפתחות הם אי-שליליים עד $3|V|+1$.

רדוקציה לקורס "מבני נתונים"

- מימוש ה-ADT יהיה בעזרת מערך בגודל $3|V|+2$ (עם אינדקסים בין 0 ל-$3V+1$).
- במקום ה- i במערך – רשימה מקושרת דו-כיוונית של מצביעים אל הקודקודים v עבורם $d[v]=i$.
- מצביעים הדדיים בין כל קודקוד בגרף והערך המתאים לו ברשימה המקושרת.

מימוש ה-ADT

- update(v, new_d)– נמחוק את המצביע לקודקוד v מהרשימה המתאימה במערך. נכניס אותו בראש הרשימה המתאימה ל-new_d. זמן: O(1).
- extract_min – נמצא את האיבר הראשון במערך שאינו מכיל רשימה ריקה.
- כיוון שכל קריאה ל- extract_min מחזירה ערך גדול או שווה לקריאה הקודמת, אין צורך לבצע את הסריקה כל פעם מתחילת המערך בזמן (O(|V|).

זמן extract_min

- הסריקה בקריאה ל-extract_min תמשיך מהמקום בו הופסקה הסריקה הקודמת.
- זאת בעזרת משתנה שיכיל את האינדקס בו הופסקה הסריקה האחרונה.
- משתנה זה יאותחל ל-0.

זמן ריצה כולל

- אמנם זמן הריצה בקריאה אחת הוא מס' התאים שנסרקו, כלומר $O(V)$, אך מכיוון שכל תא נסרק פעם אחת בכל האלגוריתם, זהו גם זמן הריצה של כל $|V|$ הקריאות ל-extract_min.
- $O(|V|+|V|*extract\_min + |E|*update) = O(|V|+|E|)$

תרגיל 2

שאלה 2 נתון גרף א-ציקלי $G=(V, E)$, $W:E \rightarrow R$, ו-$s \in V$. מצאו מסלולים קצרים ביותר מ-s לכל קודקוד.

פתרון

- נמצא מיון טופולוגי של G.
- נניח שסדר הצמתים בו הוא $u_1, .. u_k, s, v_1,.. , v_m$.
- לכל צומת u_i, $i \leq k$, נסמן $\delta(u_i)=\infty$.

האלגוריתם

- נגדיר $v_0 = s$
- נמצא מסלולים קצרים ביותר מ-s ל-v_i, $i \le m$.
- זאת בעזרת סריקה על הצמתים v_i לפי הסדר שלהם במיון הטופולוגי. בכל צומת v מבצעים relax על כל הקשתות שיוצאות ממנו (v,t).
- תזכורת ל-relax:
- $d[t]=\min\{d[t], d[v]+w((v,t))\}$.

האלגוריתם (המשך)

- איתחול: כרגיל, $d[v]=\infty$ לכל $v\in V\setminus\{s\}$, $d[s]=0$.
- זמן ריצה: $O(|E|+|V|)$ עבור המיון טופולוגי, וכנ"ל על סריקתו.

הוכחת נכונות

- **<u>טענה:</u>** כאשר מבצעים relax לקשתות שיוצאות מ-v_i, $0 \le |i| \le V$, מתקיים $d[v_i] = \delta(v_i)$.
- <u>הוכחה:</u> בעזרת אינדוקציה על i.
- עבור $i=0$ הטענה נובעת מהאיתחול.
- עבור $i>0$ נניח שהטענה נכונה לכל $j<i$.
- המשך בשקף הבא...

הוכחת הטענה כאשר אין מסלול ל-v_i

- עבור כל קשת (v_j, v_i) מתקיים $\delta(v_j)=\infty$.
- מכיוון שזוהי קשת נכנסת ל-v_i, במיון הטופולוגי מתקיים $j<i$.
- לכן, לפי הנחת האינדוקציה בזמן סריקת קשתות היוצאות מ-v_i מתקיים $d[v_j]=\delta(v_j)=\infty$.
- כיוון שערכי $d[v_j]$ יכולים רק לרדת (לפי הגדרת relax), התקיים $d[v_j]=\infty$ גם בזמן שעשינו relax לקשת (v_j, v_i).
- לכן אף קשת (v_j, v_i) לא גרמה לשינוי ערך $d[v_i]=\infty$ ההתחלתי.

הוכחת הטענה כאשר יש מסלול ל-v_i

- אם ל-v_i יש מסלול מ-$s=v_0$, אזי יש לו גם מסלול קצר ביותר.
- נניח (v_j, v_i) היא קשת אחרונה במסלול כזה.
- בגלל המיון הטופולוגי מתקיים $j<i$.

הוכחת הטענה כאשר יש מסלול ל-v_i

- מהנחת האינדוקציה מתקיים $d[v_j]=\delta\,[v_j]$ בזמן ביצוע relax לקשת (v_j, v_i), ולכן
- $d[v_i] \le d[v_j]+ w\,(v_j, v_i)= \delta\,[v_j]+ w\,(v_j, v_i)= \delta\,[v_i]$
- כמו כן, לאחר פעולת relax לקשת (v_a, v_i) ששינתה את $d[v_i]$ יתקיים
- $d[v_i]=d[v_a]+w(v_a\,, v_i)=\delta\,[v_a]+w(v_a\,, v_i) \ge\delta\,[v_i]$.
- קיבלנו ש-$d[v_i]=\delta\,[v_i]$ ברגע שנסרק v_i.

תרגיל 3

- נתון גרף מכוון $G=(V, E)$, $w:E \rightarrow R^+$, $s,t \in V$.
- כל קשת צבועה באדום או כחול.
- יש למצוא אורך מק"ב מ-s ל-t מבין המסלולים בעלי מס' זוגי של קשתות אדומות.

תשובה

- לשם נוחות, עבור צומת $v \in V$ נסמן $v_0=(v,0)$ ו- $v_1=(v,1)$.
- נגדיר $G'=(V', E')$ כדלהלן:
- $V'=\{v_0, v_1 \mid v \in V\}$, ו-
- $E'=\{(u_0, v_0), (u_1, v_1) \mid (u,v) \in E$ כחולה$\}$.
$\cup\{(u_0, v_1), (u_1, v_0) \mid (u,v) \in E$ אדומה$\}$

אלגוריתם+הוכחה

- <u>האלגוריתם:</u> הרץ Dijkstra על G’ מ-s_0. החזר את אורך המק"ב מ-s_0 ל-t_0.
- <u>הוכחת נכונות:</u> מההגדרות נובע שקיים מסלול בין s_0 ל- t_0 אםם קיים מסלול מ-s ל- t באותו אורך ובעל מס' אדומות זוגי.
- לכן מק"ב כלשהו מ-s_0 ל- t_0 הוא בעל אותו אורך של מק"ב מבין המסלולים בעלי מס' זוגי של קשתות אדומות מ-s ל-t.