יסודות מערכות תובלה ושינוע

מצגת 6

מודלים וכלים – המסלול הקצר

בעיית המסלול הקצר ביותר – Shortest Path Problem

הבעיה: מציאת המסלול הקצר ביותר (במה ?)

שימושים: לא קשה למצוא...

סוגי בעיות:

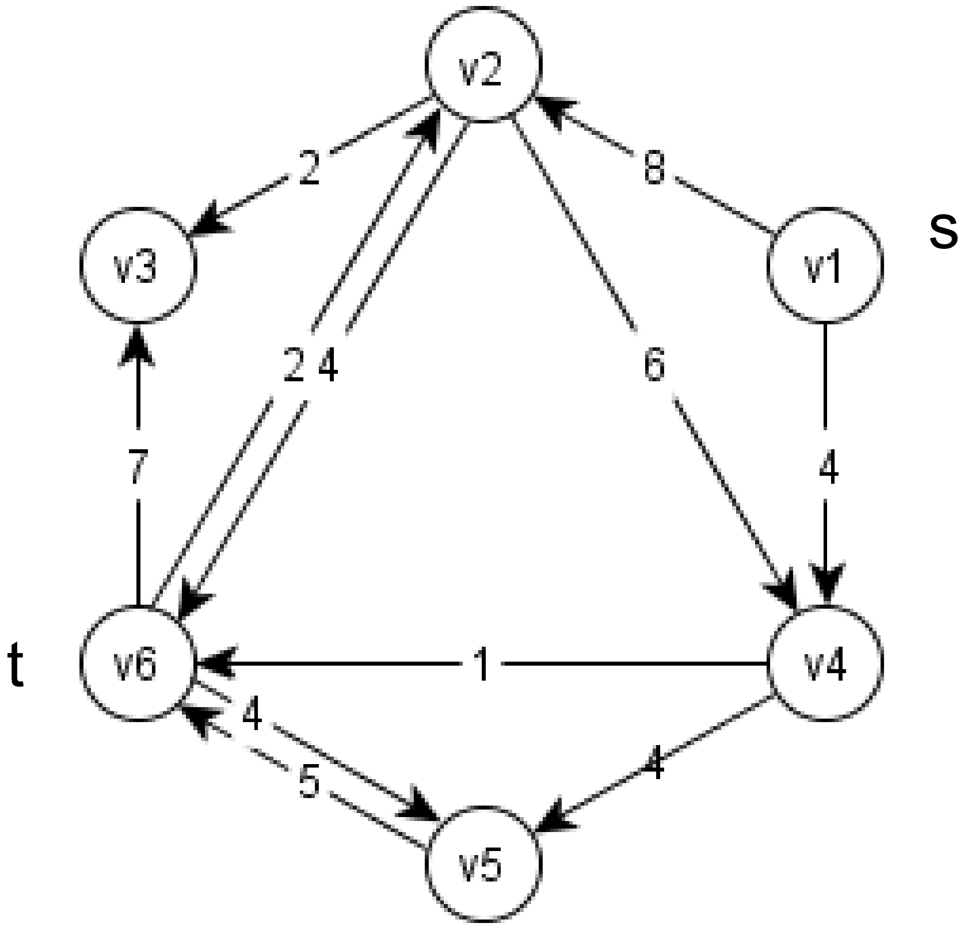
- ❑ מסלול קצר ביותר מקודקוד אחד לקודקוד אחר (לא נלמד)
- ❑ מסלול הקצר ביותר מקדקוד אחד לשאר הקודקודים
- ❑ מסלול קצר ביותר מכל קדקוד לכל קדקוד אחר
- ❑ k המסלולים הקצרים ביותר (לא נלמד)
- ❑ מסלול הקצר ביותר ברשת סטוכסטית

המסלול הקצר ביותר – אלגוריתמים

- מסלול קצר מקדקוד אחד לקדקוד אחר
  - Nicholson
- מסלול קצר ביותר מקדקוד לשאר הקדקודים
  - דיקסטרה
  - Bellman-ford, Yen (ערכים כלשהם, גילוי לולאה שלילית)
- מסלול קצר מכל הקדקודים לכל הקדקודים
  - Floyd (ערכים כלשהם, גילוי לולאה שלילית)
- K-Shortest-Path
  - Shier, Hoffman & Pavley
- מסלול קצר ברשת סטוכסטית

המסלול הקצר ביותר בין זוג קודקודים

- עבור רשת בעלת משקלים, G=(V,E,w), מהו המסלול הקצר ביותר מקודקוד נתון, s, לקודקוד נתון, t.



המסלול הקצר ביותר בין זוג קודקודים

❑ עבור רשת בעלת משקלים, G=(V,E,w), מהו המסלול הקצר ביותר מקודקוד נתון, s, לקודקוד נתון, t.

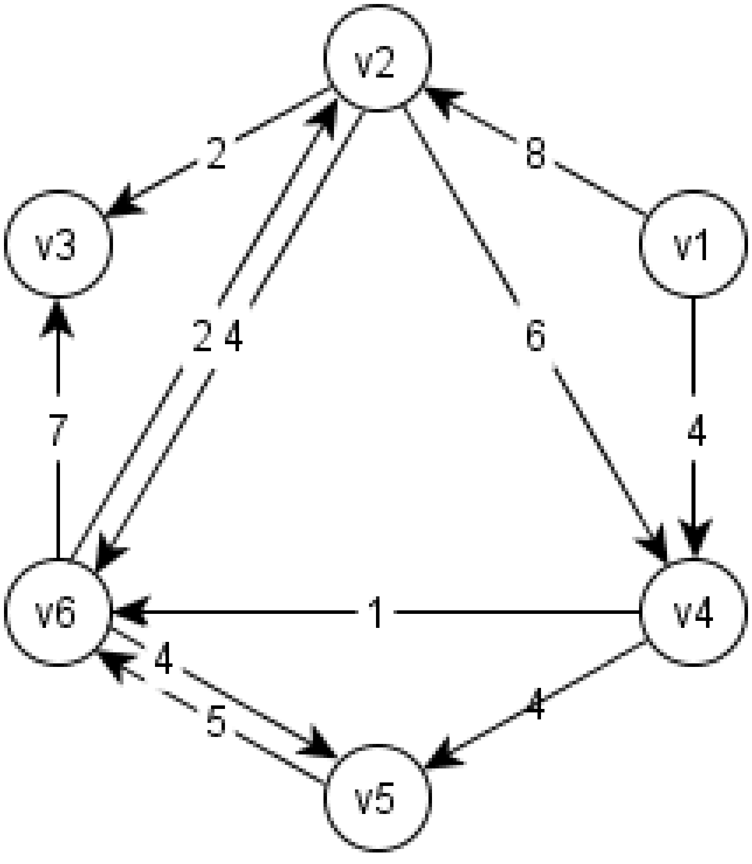
$$\min \; Z = w_{ij}x_{ij} \quad \forall (i,j) \in E$$

$$\sum_{j:(s,j)\in E} x_{sj} - \sum_{i:(i,s)\in E} x_{is} = 1$$

$$\sum_{j:(t,j)\in E} x_{tj} - \sum_{i:(i,t)\in E} x_{ti} = -1$$

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{i:(i,j)\in E} x_{ij} = 0 \quad i \neq s, j \neq t$$

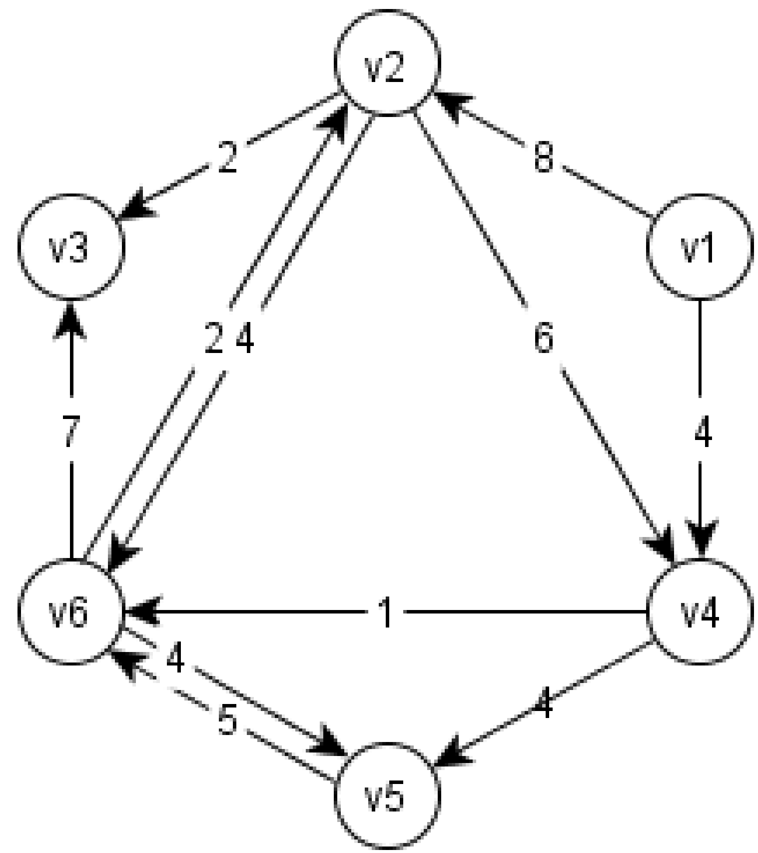
$$x_{ij} \geq 0$$



המסלול הקצר ביותר בין זוג קודקודים

- ❑ פתרון בעזרת Solver של Excel

	A	B	C	D
1				
2		i	j	w(i,j)
3		1	2	8
4		1	4	4
5		2	3	2
6		2	6	4
7		2	4	6
8		4	6	1
9		4	5	4
10		5	6	5
11		6	5	4
12		6	2	2
13		6	3	7
14				

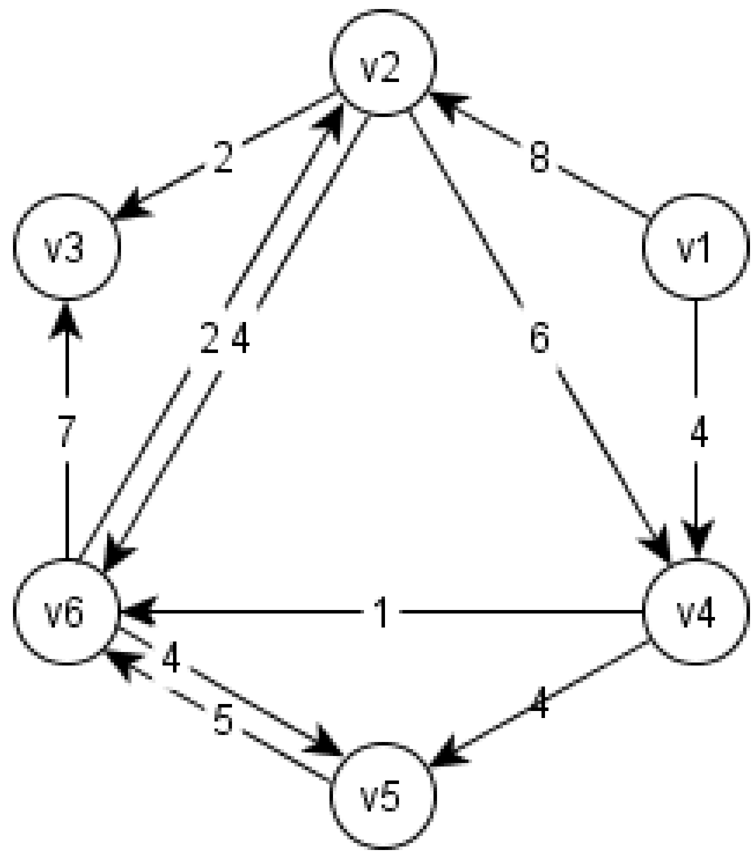


המסלול הקצר ביותר בין זוג קודקודים

- פתרון בעזרת Solver של Excel

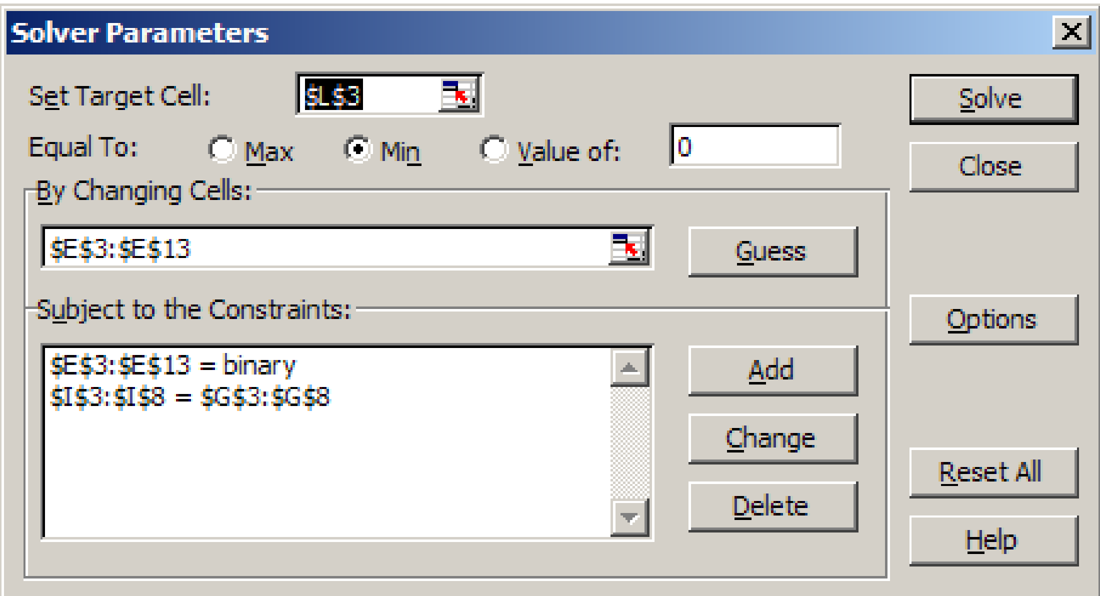
G	H	I
Constraints		
0	=	1
0	=	0
0	=	0
0	=	0
0	=	0
0	=	-1

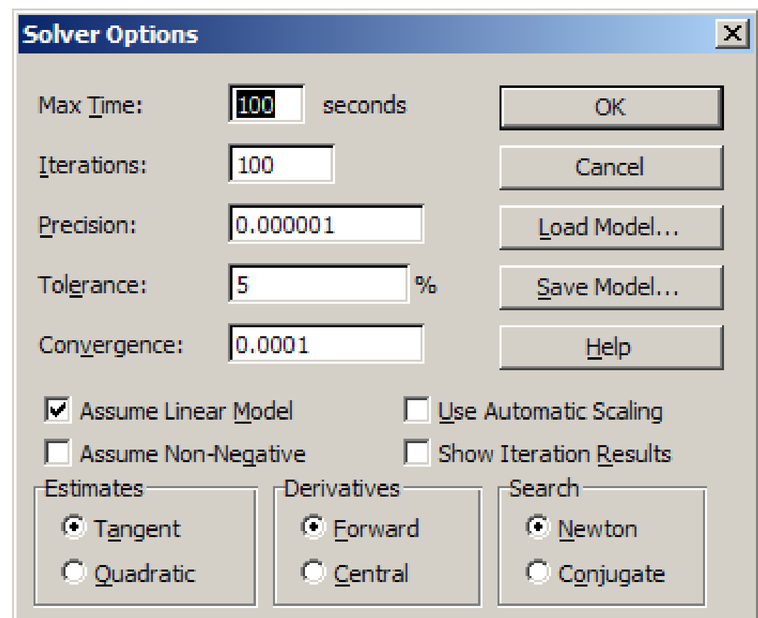
Z=		0



המסלול הקצר ביותר בין זוג קודקודים

❑ פתרון בעזרת Solver של Excel

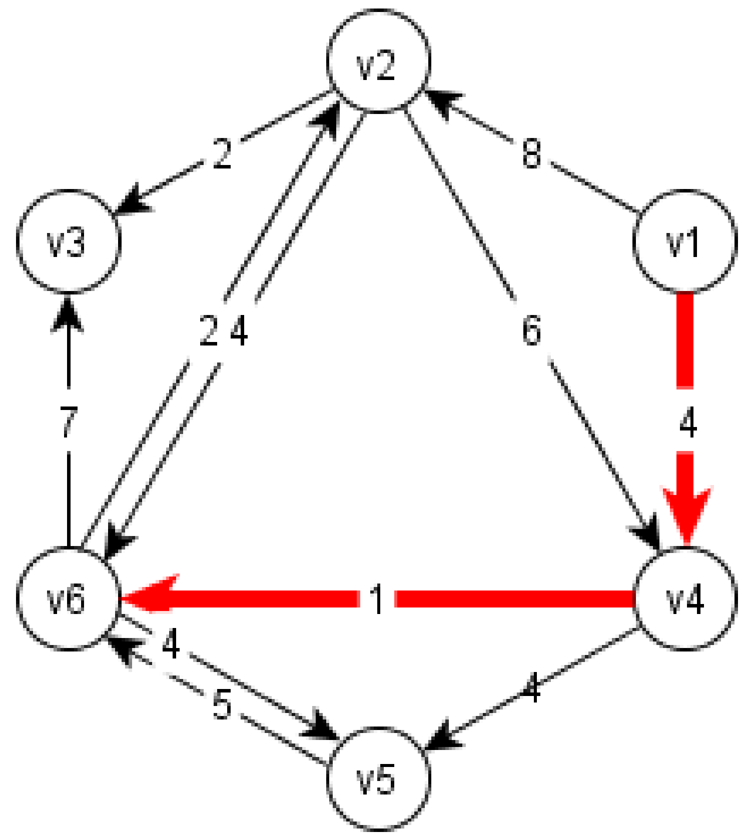




המסלול הקצר ביותר בין זוג קודקודים

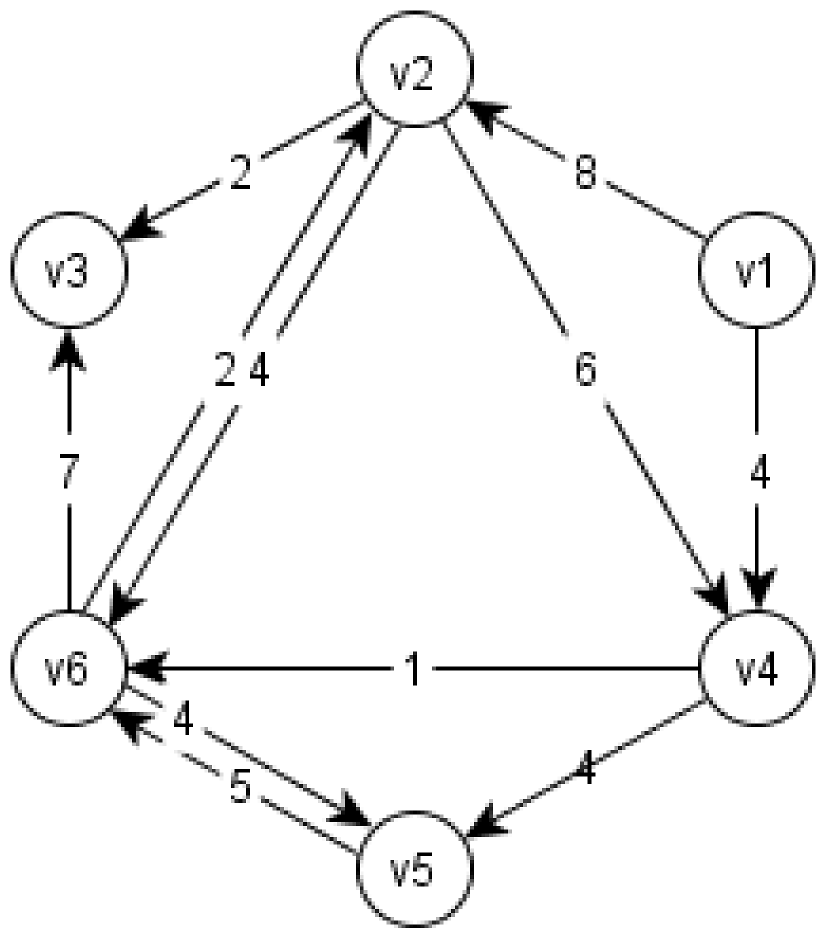
- פתרון בעזרת Solver של Excel

10					
11	Adjustable Cells				
12	Cell	Name	Original Value	Final Value	
13	E3	x(i,j)	0	4.55125E-12	
14	E4	x(i,j)	0	1	
15	E5	x(i,j)	0	0	
16	E6	x(i,j)	0	4.55125E-12	
17	E7	x(i,j)	0	0	
18	E8	x(i,j)	0	1	
19	E9	x(i,j)	0	0	
20	E10	x(i,j)	0	0	
21	E11	x(i,j)	0	0	
22	E12	x(i,j)	0	0	
23	E13	x(i,j)	0	0	
24					



המסלולים הקצרים מקודקוד לשאר הקודקודים

- עבור רשת בעלת משקלים, G=(V,E,w), מהם המסלולים הקצרים ביותר מקודקוד נתון, s, לכל שאר הקודקודים בגרף.



פתרון בעזרת תיכנות לינארי

- עבור רשת בעלת משקלים, מהם המסלולים הקצרים ביותר מקודקוד נתון, s, לכל שאר הקודקודים בגרף.

$$\text{Minimize:} \quad \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{Subject to:} \quad \sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i,)\in A\}} x_{ji} = (n-1), \quad i = s$$

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i,)\in A\}} x_{ji} = -1 \quad i = N - \{s\}$$

$$x_{ij} \geq 0 \quad (i,j) \in A$$

המסלולים הקצרים מקודקוד לשאר הקודקודים

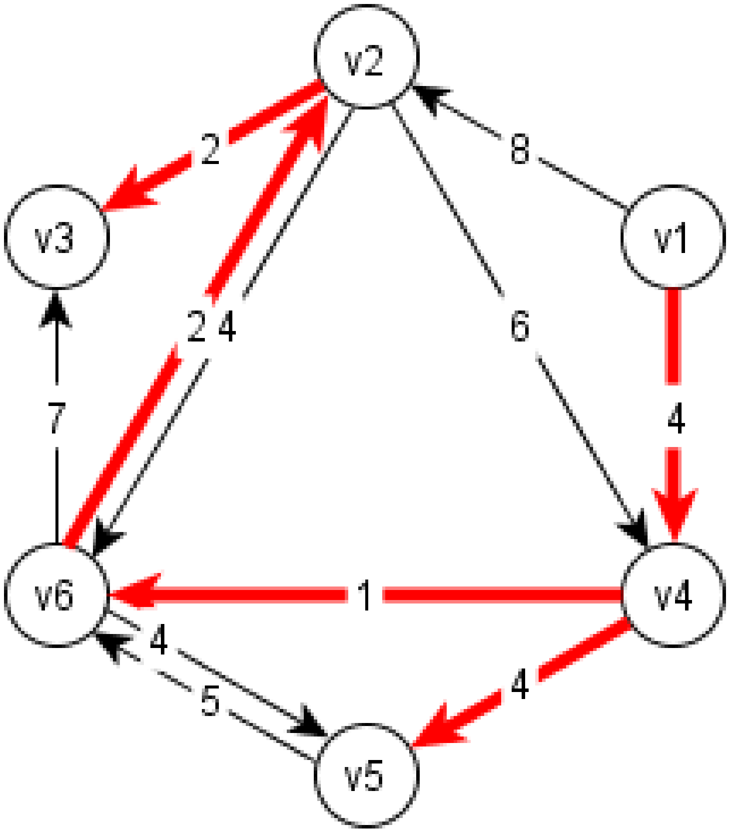
- עבור רשת בעלת משקלים, מהם המסלולים הקצרים ביותר מקודקוד נתון, s, לכל שאר הקודקודים בגרף.

Target Cell (Min)

Cell	Name	Original Value	Final Value
L3	Z=	0	33

Adjustable Cells

Cell	Name	Original Value	Final Value
E3	x(i,j)	0	0
E4	x(i,j)	0	5
E5	x(i,j)	0	1
E6	x(i,j)	0	0
E7	x(i,j)	0	0
E8	x(i,j)	0	3
E9	x(i,j)	0	1
E10	x(i,j)	0	0
E11	x(i,j)	0	0
E12	x(i,j)	0	2
E13	x(i,j)	0	0



האלגוריתם של דיקסטרה

- הוצג ע"י מדען המחשב ההולנדי אדסחר דיקסטרה ב-1959
- פותר את בעיית המסלול הקצר ביותר מקודקוד לשאר הקודקודים
- האלגוריתם פועל כאשר בגרף לא קיימות קשתות בעלות משקל שלילי
- אלגוריתם של דיקסטרה הינו אלגוריתם חמדני

האלגוריתם של דיקסטרה

- שלב האיתחול:
  - עבור כל קודקוד נסמן את המרחק אליו כ-"∞"
  - את המרחק של קודקוד המקור נסמן כ-"0"
- השלב האיטרטיבי: כל עוד קיימים קודקודים שלא ביקרנו בהם נבצע
  - מבין הקודקודים שלא ביקרנו בהם, נבחר את הקודקוד שהמרחק אליו הוא הקצר ביותר
  - נבחן את כל הקודקודים השכנים שלו
  - אם זה קודקוד שכן שלא הגענו אליו קודם לכן, נציין מאיזה קודקוד הגענו, ומה המרחק שלקח להגיע אליו
  - אם כבר הגענו לקודקוד השכן הזה, אך המרחק החדש קטן יותר, נתקן את נתוני הקודקוד בהתאם

האלגוריתם של דיקסטרה - דוגמא

v1 0

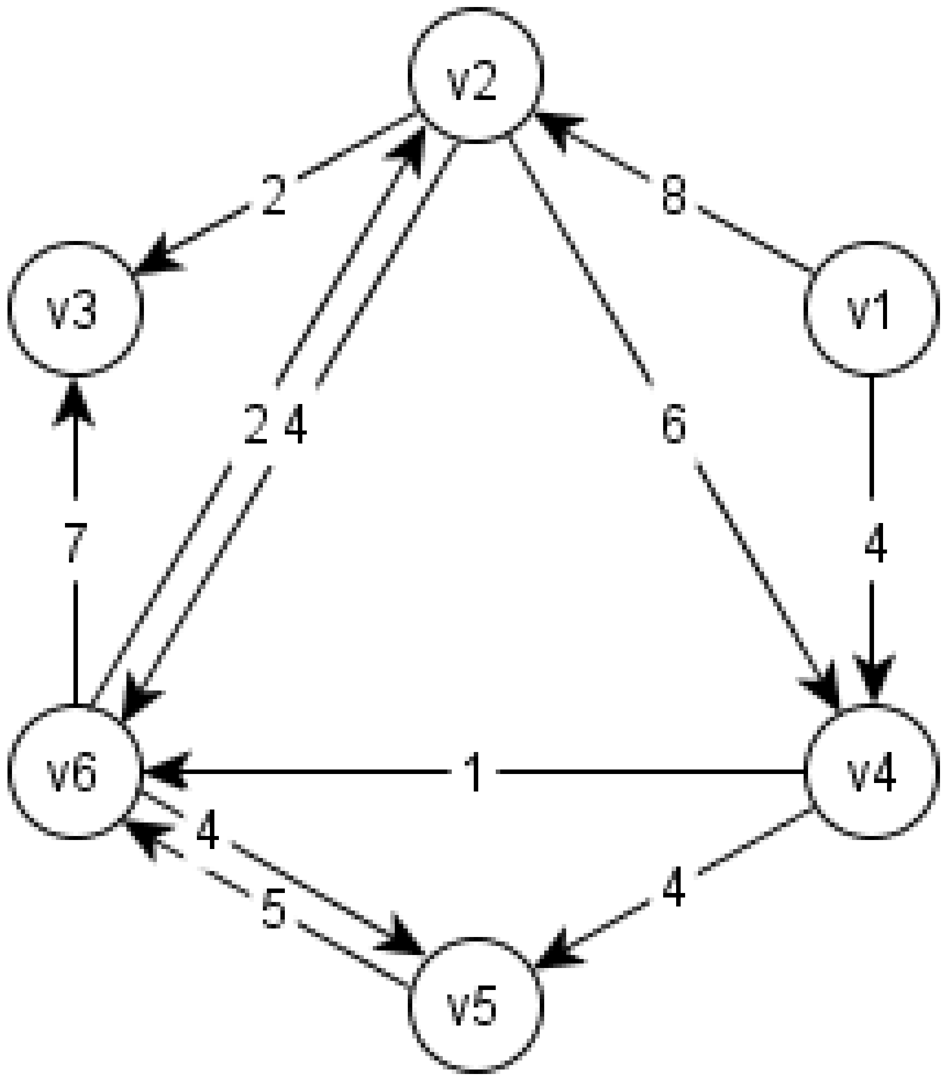
v2 ∞

v3 ∞

v4 ∞

v5 ∞

v6 ∞



האלגוריתם של דיקסטרה - דוגמא

v1 0

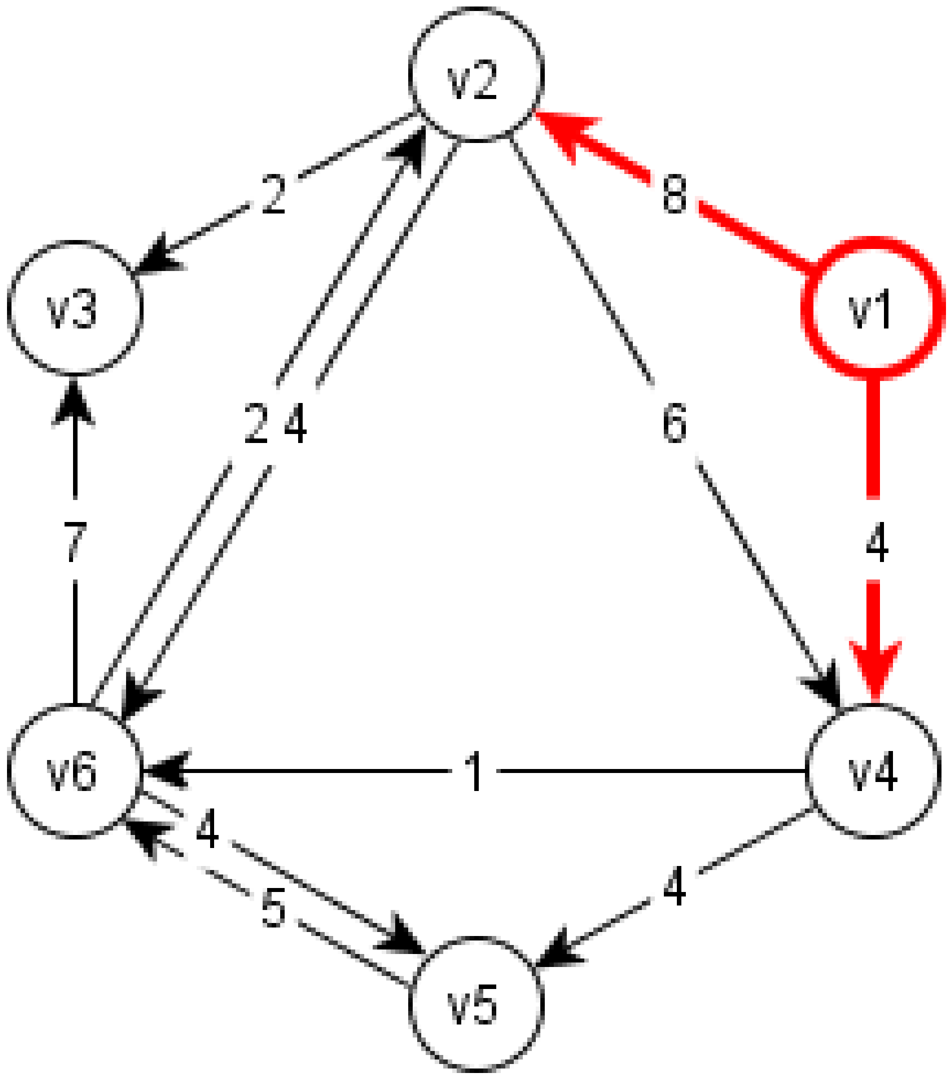
v2 8

v3 ∞

v4 4

v5 ∞

v6 ∞



האלגוריתם של דיקסטרה - דוגמא

v1 0

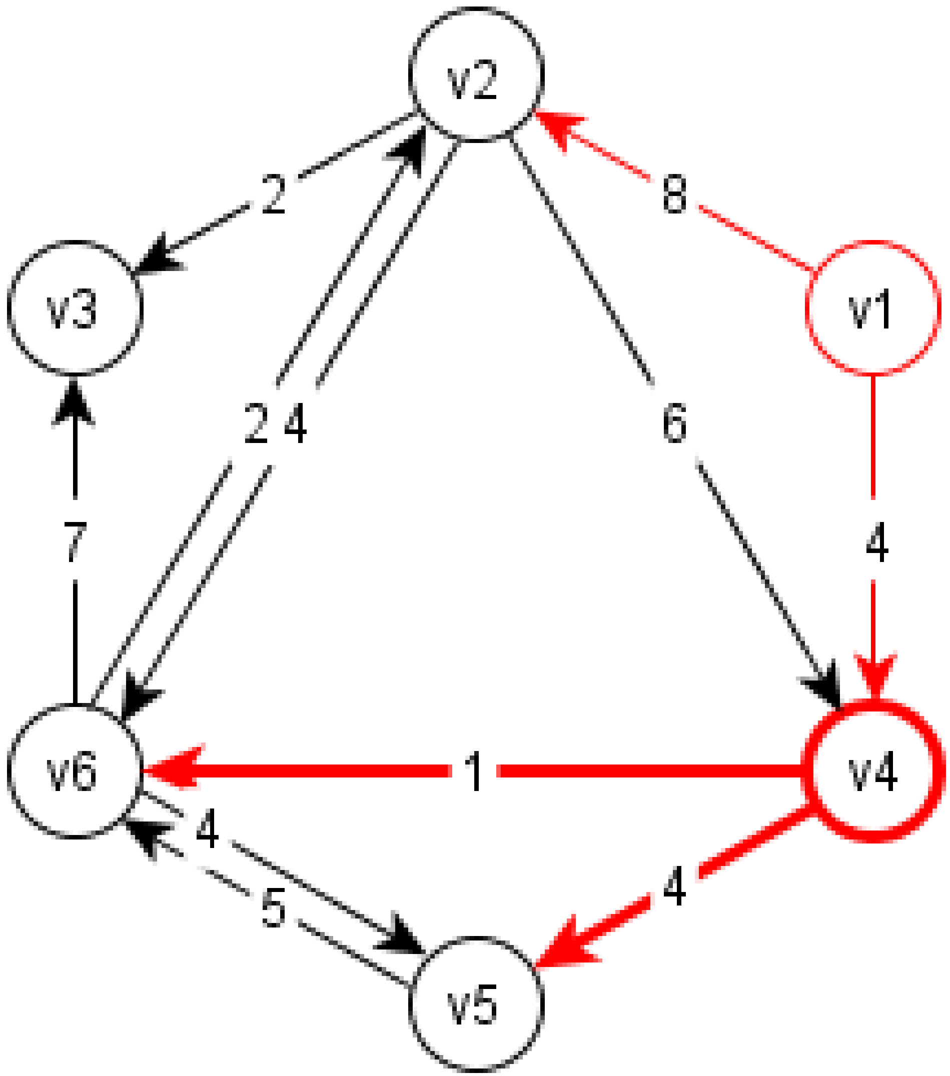
v2 8

v3 ∞

v4 4

v5 8

v6 5



האלגוריתם של דיקסטרה - דוגמא

v1 0

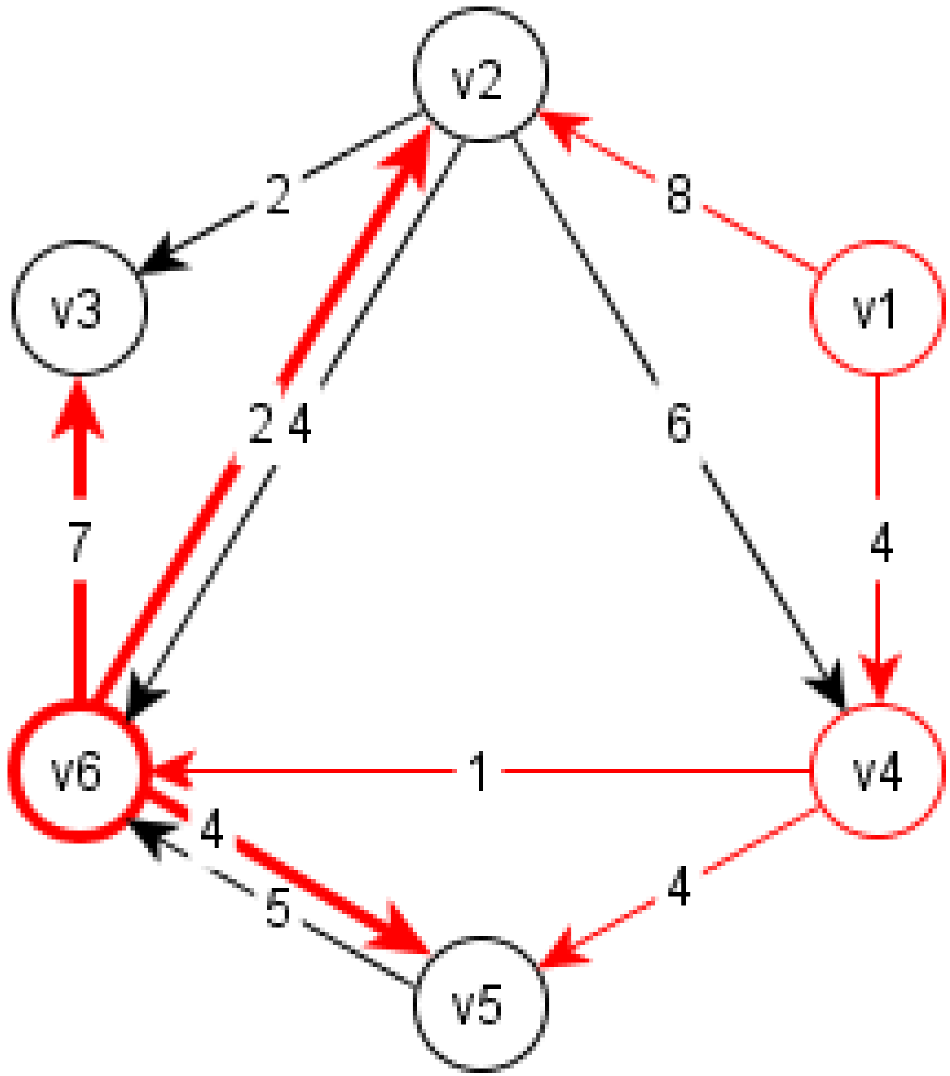
v2 7

v3 12

v4 4

v5 8

v6 5



האלגוריתם של דיקסטרה - דוגמא

v1 0

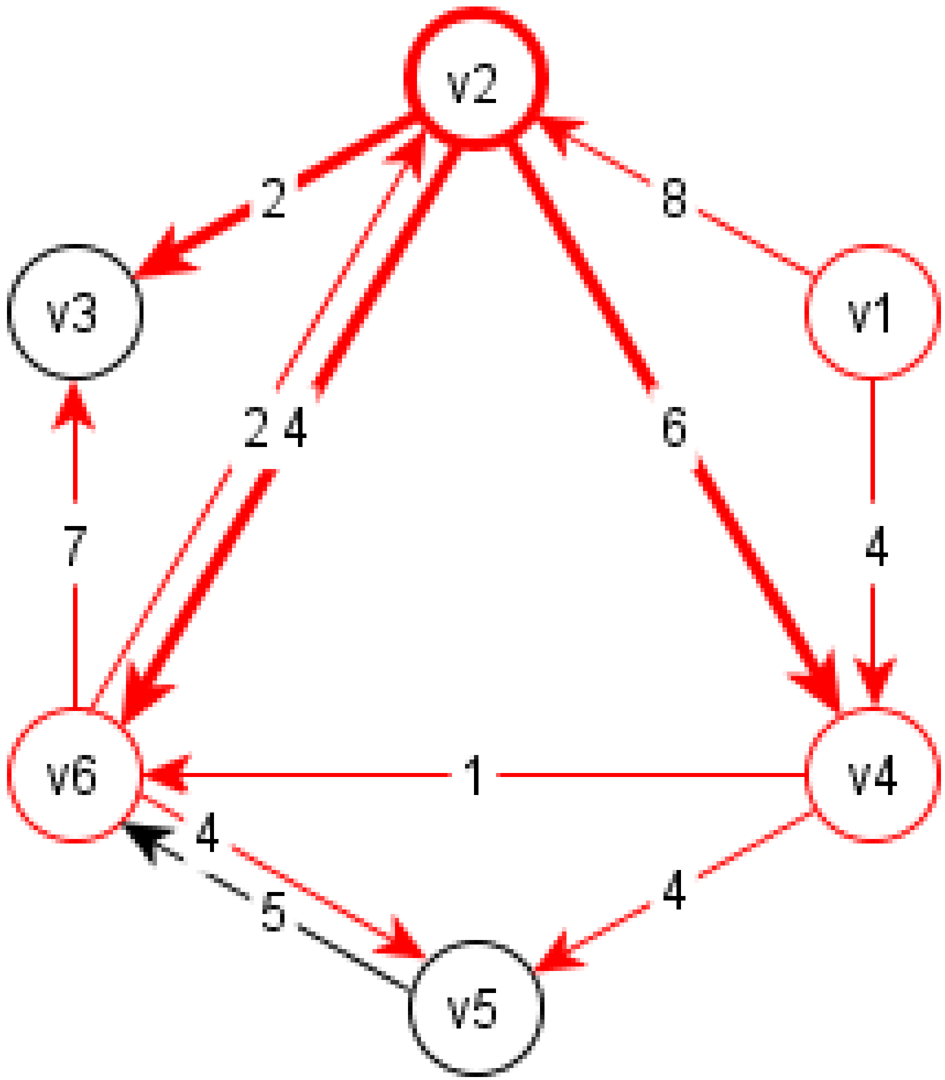
v2 7

v3 9

v4 4

v5 8

v6 5



האלגוריתם של דיקסטרה - דוגמא

v1 0

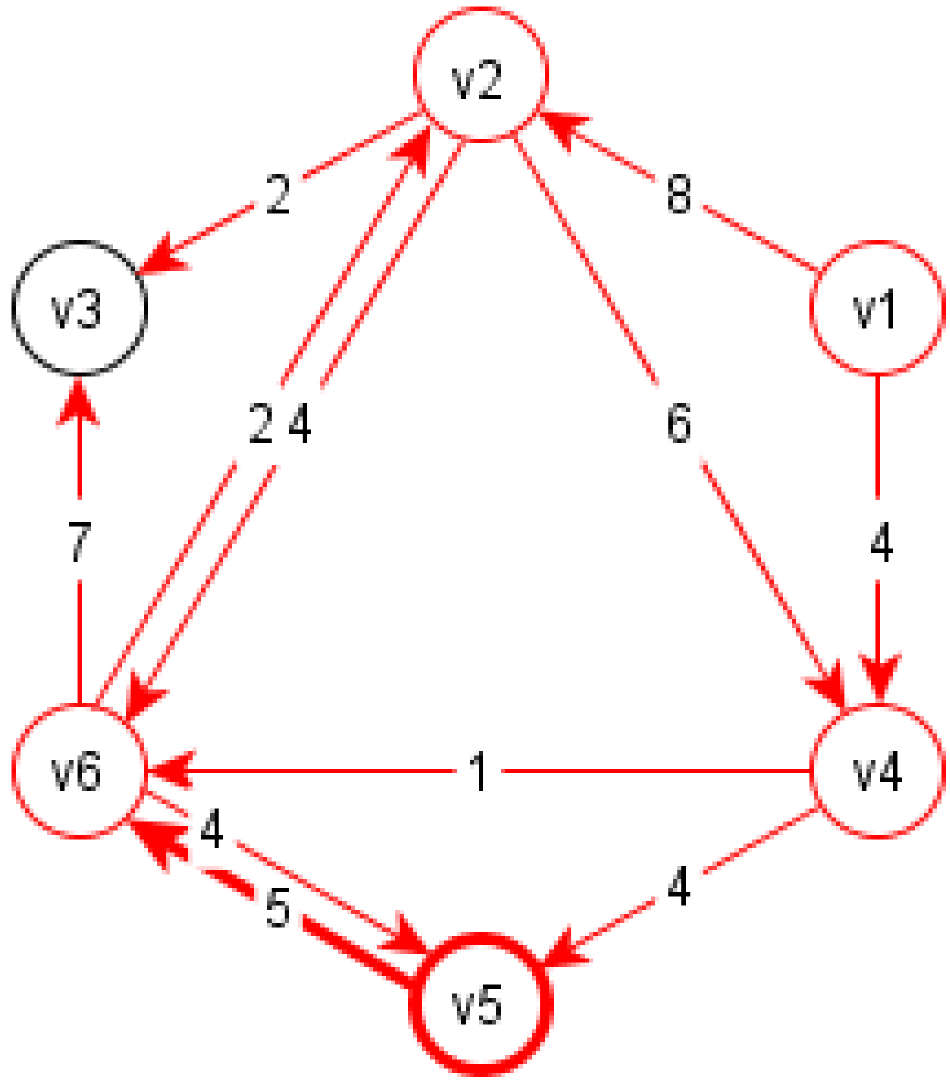
v2 7

v3 9

v4 4

v5 8

v6 5



האלגוריתם של דיקסטרה - דוגמא

v1 0

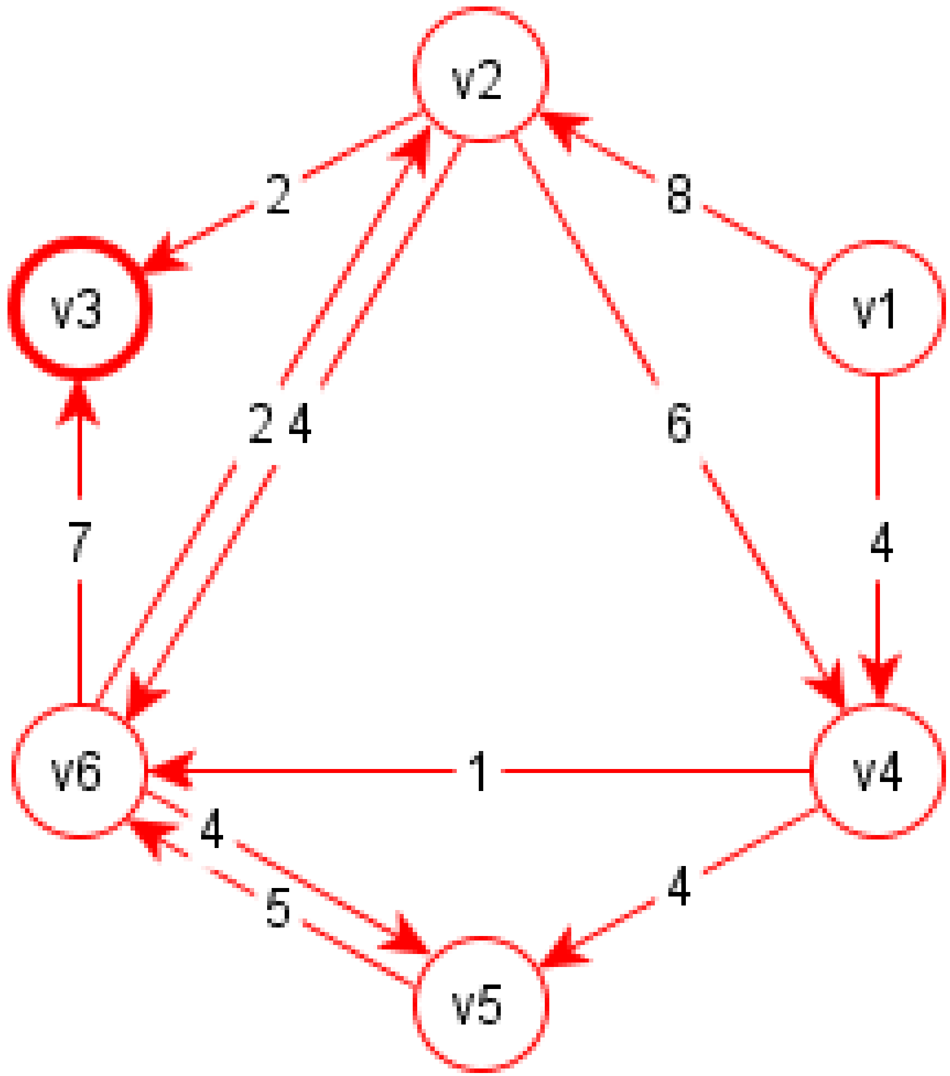
v2 7

v3 9

v4 4

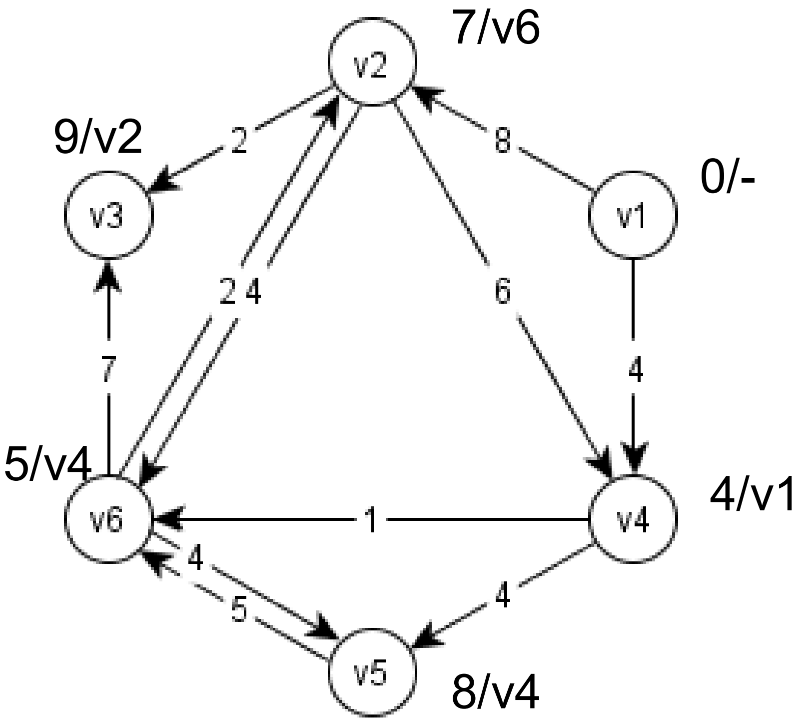
v5 8

v6 5



האלגוריתם של דיקסטרה - דוגמא

V1-V2	7	V1-V4-V6-V2
V1-V3	9	V1-V4-V6-V2-V3
V1-V4	4	V1-V4
V1-V5	8	V1-V4-V5
V1-V6	5	V1-V4-V6



נכונות האלגוריתם של דיקסטרה

```
Dijkstra(G,s)
for all u in V-s set d(u)=∞
d(s)=0
R={}
While R ≠ V
        pick u from V, u is not in R with smallest d(u)
        R=R+u
        for all vertices v adjacent to u
                if d(v)>d(u)+Length(u,v)
                        d(v)=d(u)+Length(u,v)
```

נכונות האלגוריתם של דיקסטרה

- המטרה היא להראות שכל x ששייך ל-R, d(x)=D(x), כאשר D(x) הוא המרחק הקצר ביותר מ-s ל-X
- נוכיח באמצעות אינדוקצייה
- מקרה ראשוני - R מכיל רק איבר 1. מכוון ש-R רק גדל, הפעם היחידה שבה R מכיל רק איבר אחד היא כאשר R={s}, מכוון ש-d(s)=0 וגם D(s)=0 הרי שבמקרה זה הטענה נכונה
- הנחת האינדוקצייה – כאשר R מכיל k קודקודים (או פחות), אז לכל קודקוד x ששייך ל-R מתקיים d(x)=D(x)
- יש להראות שעבור R'=R+u, כש-u הוא קודקוד כלשהו ששייך לקבוצה V-R, הנחת האינדוקצייה גם כן מתקיימת

נכונות האלגוריתם של דיקסטרה

- נסמן ב-P את המסלול מ-s ל-u בעל המרחק d(u)
- המסלול P מורכב מהמסלול P’ (שהינו המסלול מ-s ל-v) ומהקשת (v,u)
- נניח שיש מסלול קצר יותר מ-s ל-u, שהמרחק שלו הוא D(u), נסמן אותו ב-Q
- נסמן ב-y את הקודקוד הראשון במסלול Q שאינו שייך לקבוצה R
- במסלול הזה חייבת להיות קשת (x,y), כשהקודקוד x שייך לקבוצה R
- ניתן לחלק את המסלול Q לשלושה תתי-מסלולים באופן הבא: Q_1 (שהינו מסלול מ-s ל-x), הקשת (x,y) ו-Q_2 (שהינו מסלול מ-y ל-u)
- על פי הנחת האינדוקציה d(x)=D(X)
- בהתאם לאופן שבו האלגוריתם מעדכן את מרחקי הקודקודים $d(y) \le d(x) + Length(x,y)$

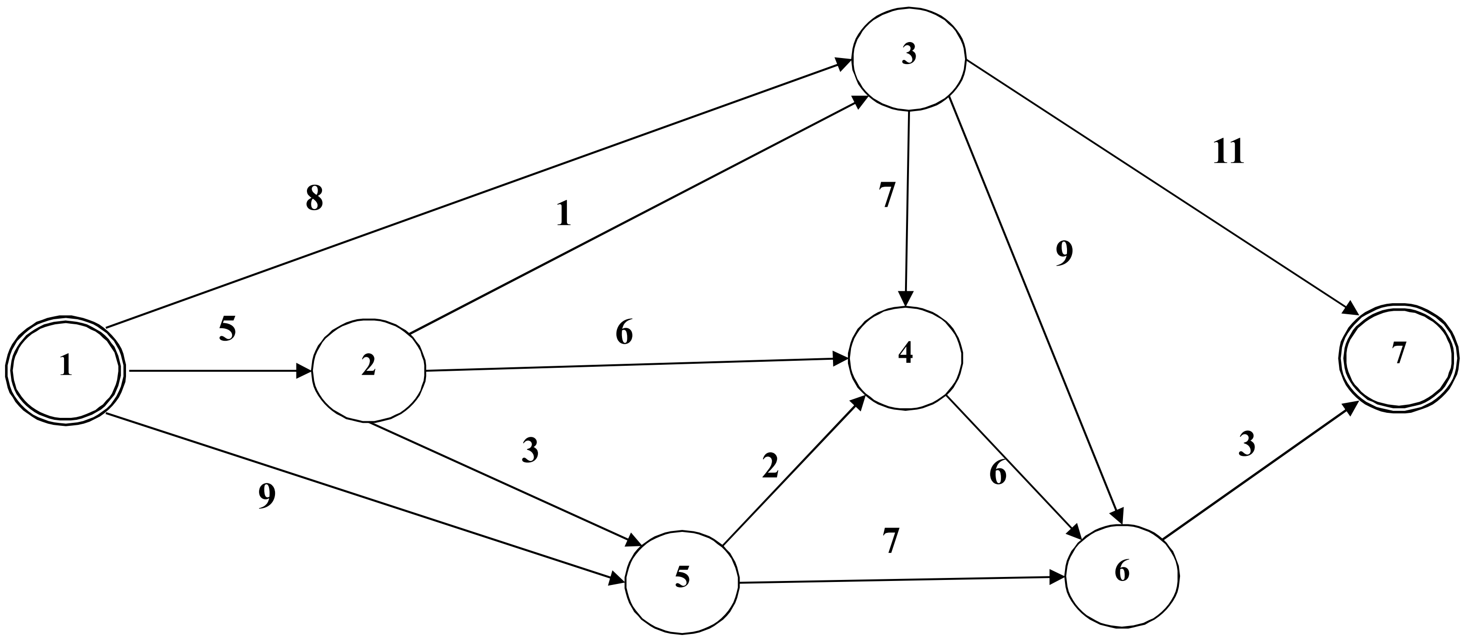
נכונות האלגוריתם של דיקסטרה

- בשל האופי החמדני של האלגוריתם $d(u) \le d(y)$ (מכוון שלקבוצה R אנו מוסיפים קודקודים לפי סדר המרחקים)
- אבל, $d(u) \le d(y) + d(Q_2)$, ומכאן נובע ש-$d(Q_2)=0$
- ולכן לא קיים מסלול Q שהוא קצר מ-P

סיבוכיות אלגוריתם של דיקסטרה

- בתחילת האלגוריתם יש לנו גרף ריק
- בכל שלב אנו בוחנים קודקוד אחד, ואת הקשתות שלו
- בסוף האלגוריתם עברנו על כל הקודקודים של הגרף
- סיבוכיות האלגוריתם עבור הגרף, $G=(V,E)$, היא $\boldsymbol{O}(|V|)$

האלגוריתם של דיקסטרה – דוגמא נוספת



אלגוריתם פלויד-וורשאל

- האלגוריתם משמש למציאת המסלולים הקצרים ביותר בין כל שני זוגות קודקודים בגרף ממושקל ומכוון.
- האלגוריתם פועל גם כאשר בגרף קיימות קשתות בעלות משקל שלילי
- האלגוריתם אינו פועל אם בגרף יש מעגל שלילי.
- סיבוכיות האלגוריתם היא $O(|V|^3)$

אלגוריתם פלויד-וורשאל

מרחקי דרך בישראל

	אילת	אשדוד	אשקלון	באר שבע	בית שאן	זכרון יעקב	חדרה	חיפה	טבריה	ירושלים	לוד (נתב"ג)	מטולה	נהריה	נצרת	נתניה	סדום	עין גדי	עפולה	ערד	פתח תקוה	צפת	רחובות	רמלה
אילת																							
אשדוד	333																						
אשקלון	306	28																					
באר שבע	246	82	61																				
בית שאן	378	156	171	207																			
זכרון יעקב	414	107	122	168	74																		
חדרה	392	85	100	146	71	23																	
חיפה	439	132	147	193	71	34	49																
טבריה	416	170	185	231	38	79	85	65															
ירושלים	323	71	74	89	122	119	98	145	160														
לוד (נתב"ג)	342	43	58	97	124	75	53	100	108	47													
מטולה	480	233	248	294	102	142	148	118	64	224	201												
נהריה	472	168	183	229	94	67	85	36	63	181	136	85											
נצרת	417	142	157	203	39	47	57	40	32	142	110	95	55										
נתניה	381	74	89	135	89	40	18	64	103	88	44	166	100	75									
סדום	186	167	146	85	192	253	231	264	230	137	181	294	287	232	220								
עין גדי	240	153	157	121	138	202	180	209	176	83	129	240	232	177	171	55							
עפולה	405	129	144	190	27	47	44	45	41	129	97	104	68	13	62	219	165						
ערד	221	131	111	50	209	217	195	242	247	111	146	311	278	248	185	35	72	236					
פתח תקוה	347	47	66	106	109	58	40	91	124	60	10	88	123	94	30	187	97	83	151				
צפת	450	197	212	258	72	102	112	78	34	194	165	48	52	59	130	264	210	68	281	160			
רחובות	327	24	39	82	135	86	65	111	149	59	20	212	147	121	54	166	142	108	131	29	176		
רמלה	333	34	49	87	131	82	61	107	145	48	10	208	143	117	51	172	130	104	136	18	172	12	
תל אביב	349	42	57	103	120	71	49	95	134	62	15	196	131	106	33	188	143	93	152	10	160	21	20

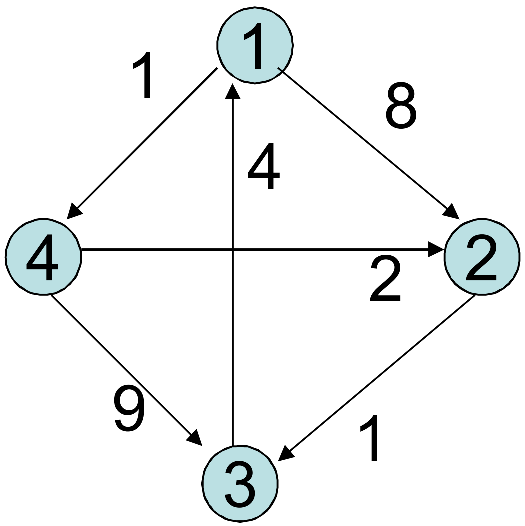
אלגוריתם פלויד-וורשאל

- לגרף נתון $G=(V,E,w)$ קיימת מטריצת מרחקים D. המרחק בין זוג קודקודים x ו-y מצויין כ-$d(x,y)$. אם לא קיימת קשת בין x ל-y, אז $d(x,y)=\infty$.
- המרחק מקודקוד לעצמו הוא $d(x,x)=0$ (אבל אם חייבים לצאת מהקודקוד (אסור להשאר במקום), אז המרחק הוא $d(x,x)=\infty$).
- האלגוריתם מבוסס על ההנחה הבאה: עבור כל זוג קודקודים x, ו-y המרחק הקצר ביותר בניהם יכול להיות
- $d(x,y)$
- או שהוא יכול לעבור דרך קודקוד נוסף, k, ואז המרחק הוא $d(x,k)+d(k,y)$.

אלגוריתם פלויד-וורשאל

1. For k=1 to |V|
2. For i=1 to |V|
3. For j=1 to |V|
4. $d_k(i,j)=\min\{d_{k-1}(i,j),d_{k-1}(i,k)+d_{k-1}(k,j)\}$

אלגוריתם פלויד-וורשאל – דוגמא 1



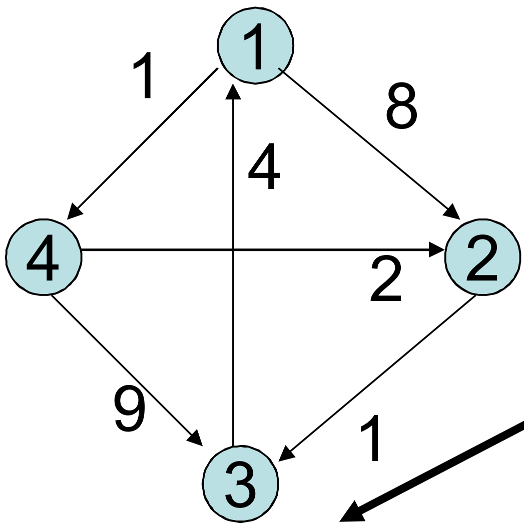
k=

$$\min\{d_{k-1}(i,j),d_{k-1}(i,k)+d_{k-1}(k,j)\}$$

i\j	1	2	3	4
1				
2				
3				
4				

i\j	1	2	3	4
1				
2				
3				
4				

אלגוריתם פלויד-וורשאל – דוגמא 1



$k=V_1$

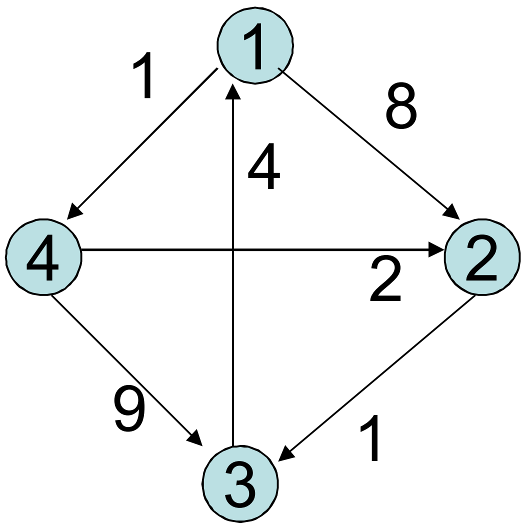
$$\min\{d_{k-1}(i,j),d_{k-1}(i,k)+d_{k-1}(k,j)\}$$

i\j	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	∞	0	∞
4	∞	2	9	0

i\j	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	9	0

1,1+1,1	0+0=0
1,1+1,2	0+8=8
1,1+1,3	0+∞=∞
1,1+1,4	0+1=1
2,1+1,1	∞+0=∞
2,1+1,2	∞+8=∞
2,1+1,3	∞+∞=∞
2,1+1,4	∞+1=∞
3,1+1,1	4+0=4
3,1+1,2	4+8=12
3,1+1,3	4+∞=∞
3,1+1,4	4+1=5
4,1+1,1	∞+0=∞
4,1+1,2	∞+8=∞
4,1+1,3	∞+∞=∞
4,1+1,4	∞+1=∞

אלגוריתם פלויד-וורשאל – דוגמא 1



$k=V_2$

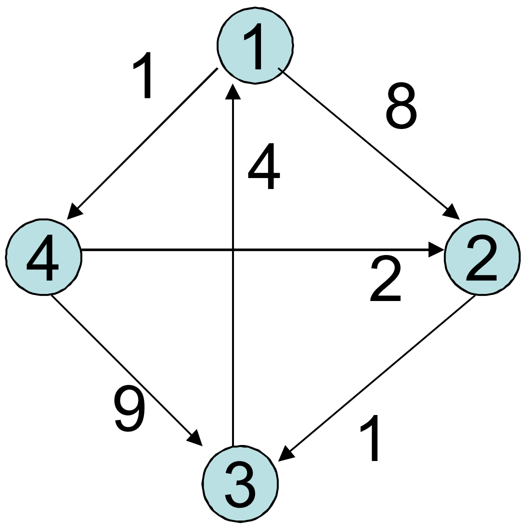
$\min\{d_{k-1}(i,j),d_{k-1}(i,k)+d_{k-1}(k,j)\}$

	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	9	0

i\j	1	2	3	4
1	0	8	9	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	3	0

1,2+2,1	8+∞=∞
1,2+2,2	8+0=8
1,2+2,3	8+1=9
1,2+2,4	-
2,2+2,1	-
2,2+2,2	0+0=0
2,2+2,3	0+1=1
2,2+2,4	-
3,2+2,1	-
3,2+2,2	12+0=12
3,2+2,3	12+1=13
3,2+2,4	-
4,2+2,1	-
4,2+2,2	2+0=2
4,2+2,3	2+1=3
4,2+2,4	-

אלגוריתם פלויד-וורשאל – דוגמא 1



$k=V_3$

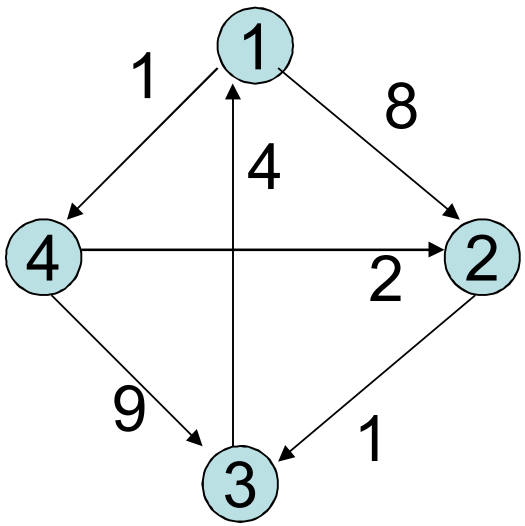
$$\min\{d_{k-1}(i,j),d_{k-1}(i,k)+d_{k-1}(k,j)\}$$

	1	2	3	4
1	0	8	9	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	3	0

i\j	1	2	3	4
1	0	8	9	1
2	5	0	1	6
3	4	12	0	5
4	7	2	3	0

1,3+3,1	9+4=13
1,3+3,2	9+12=21
1,3+3,3	9+0=9
1,3+3,4	9+5=14
2,3+3,1	1+4=5
2,3+3,2	1+12=13
2,3+3,3	1+0=1
2,3+3,4	1+5=6
3,3+3,1	0+4=4
3,3+3,2	0+12=12
3,3+3,3	0+0=0
3,3+3,4	0+5=5
4,3+3,1	3+4=7
4,3+3,2	3+12=15
4,3+3,3	3+0=3
4,3+3,4	3+5=8

אלגוריתם פלויד-וורשאל – דוגמא 1



	1	2	3	4
1	0	8	9	1
2	5	0	1	6
3	4	12	0	5
4	7	2	3	0

$k=V_4$

$\min\{d_{k-1}(i,j),d_{k-1}(i,k)+d_{k-1}(k,j)\}$

i\j	1	2	3	4
1	0	3	4	1
2	5	0	1	6
3	4	7	0	5
4	7	2	3	0

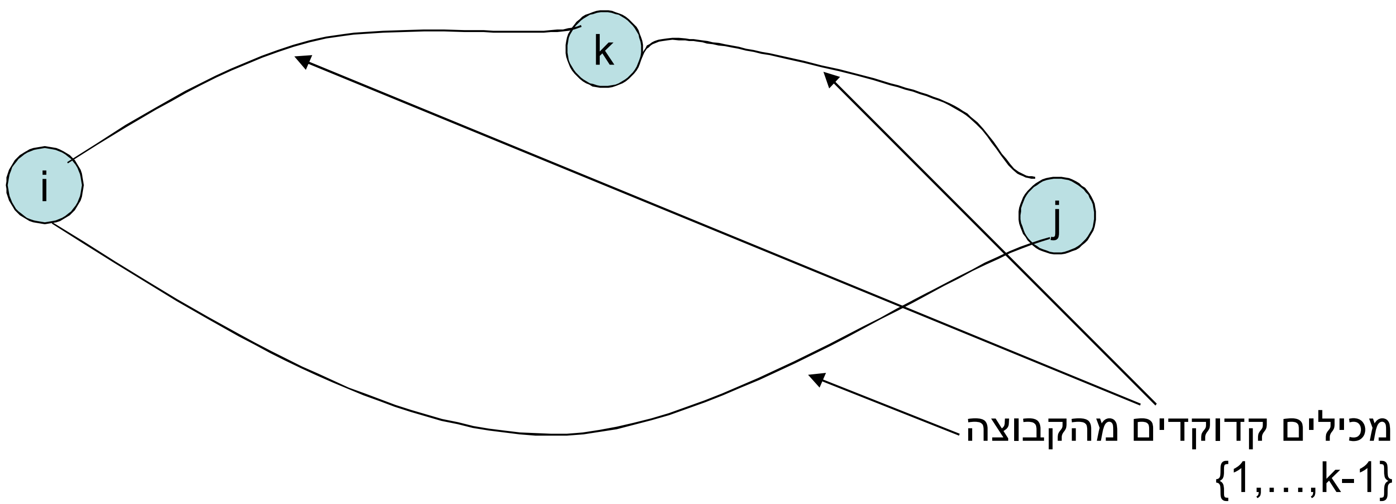
1,4+4,1	1+7=8
1,4+4,2	1+2=3
1,4+4,3	1+3=4
1,4+4,4	1+0=1
2,4+4,1	-
2,4+4,2	6+2=8
2,4+4,3	-
2,4+4,4	-
3,4+4,1	5+0=5
3,4+4,2	5+2=7
3,4+4,3	5+3=8
3,4+4,4	5+0=5
4,4+4,1	0+7=7
4,4+4,2	0+2=2
4,4+4,3	0+3=3
4,4+4,4	0+0=0

נכונות אלגוריתם פלויד-וורשאל

- הוכחה באינדוקציה
- נניח שלפני איטרציה k מתקיים שעבור כל זוג קודקודים i ו-j, התא המתאים במצטריצה, Yij, מכיל את המרחק הקצר ביותר, Q, מ-i ל-j שמכיל צמתי מעבר אפשריים ששייכים לקבוצת הקודקודים {1,…,k-1}.
- הנחה זו מתקיימת עבור שלב האיתחול (שבו קבוצת צמתי המעבר האפשריים היא קבוצה ריקה).

נכונות אלגוריתם פלויד-וורשאל

- באיטרציה k, אנו משווים את האורך של Q לאורך של המסלול R, שמורכב משני תתי מסלולים R1 ו-R2.
- R1 הוא מסלול מ-i ל-k עם צמתי מעבר אפשריים ששייכים לקבוצת הקודקודים {1,…,k-1}. R2 הוא מסלול מ-k ל-ח עם צמתי מעבר אפשריים ששייכים אף הם לקבוצת הקודקודים {1,…,k-1}.

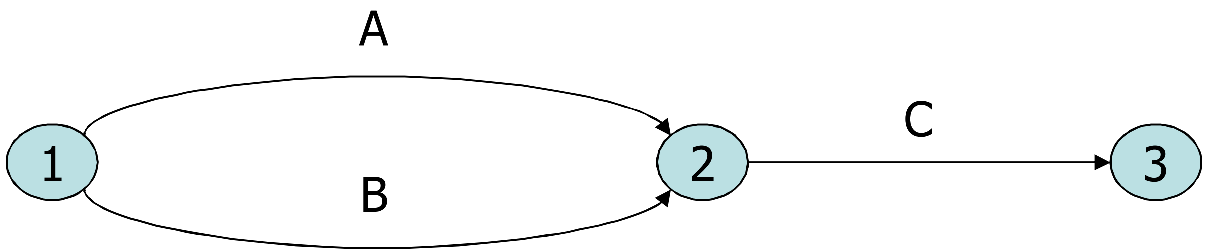


נכונות אלגוריתם פלויד-וורשאל

- המסלול הקצר ביותר מ-i ל-j בגרף G יכול להכיל צמתי מעבר אפשריים ששייכים לקבוצת הקודקודים {1,…,k}. שני מקרים יכולים להתקיים:
  - המסלול הקצר ביותר מ-i ל-j בגרף G אינו עובר דרך קודקוד k, ולכן זהו המסלול שקיבלנו באיטרצייה הקודמת.
  - המסלול הקצר ביותר מ-i ל-j בגרף G עובר דרך קודקוד k, ולכן אפשר לחלק אותו לשני מסלולים, האחד מ-i ל-k והשני מ-k ל-j, שאת שניהם קיבלנו באיטרצייה הקודמת.
- ולכן, תהליך העידכון מבטיח את הנחת האינדוקצייה לאחר איטרצייה k.

מסלול קצר ביותר ברשת סטוכסטית

- איך פותרים את הבעיה הבאה:

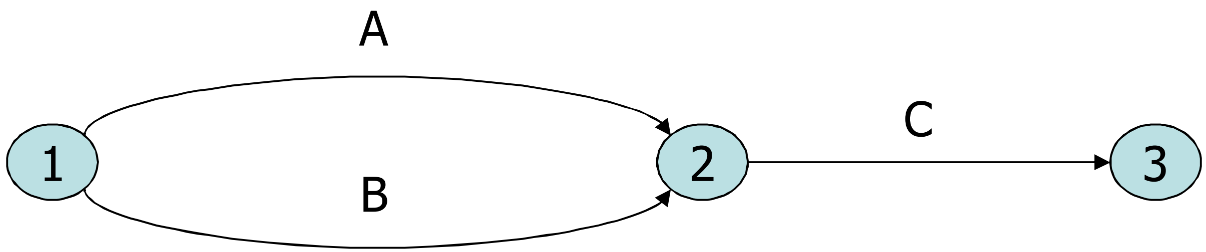


- נוסע יוצא בשעה 14:00 מקודקוד 1, ומעוניין להגיע לקודקוד 3.
- מהוא המסלול הקצר ביותר ?

<table dir="rtl">
<tr><th colspan="2">מאפייני הרשת</th></tr>
<tr><td>קשת</td><td>זמן נסיעה (דקות)</td></tr>
<tr><td>A</td><td>100</td></tr>
<tr><td>B</td><td>90 בהסתברות 0.5
120 בהסתברות 0.5</td></tr>
<tr><td>C</td><td>30 אם מגיעים ל-2 לפני 15:35
100 אחרת</td></tr>
</table>

מסלול קצר ביותר ברשת סטוכסטית

- נבחן את הפתרונות האפשריים:



- המסלול A-C: 100 + 100 = 200.
- מסלול B-C אפשרות 1: 90 + 30 = 120 בהסתברות 0.5.
- מסלול B-C אפשרות 2: 120 + 100 = 220 בהסתברות 0.5.
- האם יש חשיבות קריטית לזמן ההגעה ?