

מגיש:
מיכאל לוגסי
305536575

אלגוריתמים מבוזרים תרגיל 1

שאלה 1:

סעיף א:

צומת parent הוא הצומת הראשון שממנו התקבלה הודעת קישור (למשל "CONNECT" לצומת הנוכחי. הצומת הנוכחי מאמץ אותו כצומת הורה בעץ הפורש ומשיב לו "ACK".

צומת child הוא כל צומת שכנה אשר שלחה הודעת "ACK" בחזרה לצומת הנוכחי, בעקבות כך שהוא שלח אליה בקשת "CONNECT" כלומר, כל שכן שקיבל את הצומת הנוכחי כהורה בעץ, נחשב ל-child שלו.

כלומר, קשרי הורה-ילד נוצרים באופן חד-כיווני על בסיס קבלת הודעת "CONNECT" ראשונה, ו־ACK חזרה, תוך שמירה על כך שלא ייווצרו מחזורים.

סעיף ג':

סיבוכיות הודעות היא - $O(m)$ - כאשר m הוא מספר הקשתות בגרף (בכל קשת עוברת הודעה אחת של "CONNECT" ואחת נוספת של "ACK" או "NACK" זמן הריצה הוא, $O(D)$ כאשר D הוא הקוטר של הגרף כלומר, המספר המקסימלי של קפיצות בין שורש (root) לצומת כלשהו בגרף

```
class Node:

    def __init__(self, name: str, neighbors: list[str]):

        self.Name = name

        self.Neighbors = neighbors

        self.Parent = None

        self.Childs = []

        self.Got_rsp = {neighbor: False for neighbor in neighbors}


    def send_msg(self, msg: str, dest: str):

        # sending a message to dest


    def build(self):

        self.Parent = None

        for node in self.Neighbors:

            self.send_msg("CONNECT", node)


    def on_receive_msg(self, msg: str, src: str):

        if msg == "CONNECT":

            self.on_receive_connect(src)

        elif msg == "ACK":

            self.on_receive_ack(src)

        elif msg == "NACK":

            self.on_receive_nack(src)
```

```

def on_receive_connect(self, src: str):
    self.Got_rsp[src] = True
    if self.Parent is None:
        self.Parent = src
        for node in self.Neighbors:
            if node != src:
                self.send_msg("CONNECT", node)
    else:
        self.send_msg("NACK", src)

def on_receive_ack(self, src: str):
    self.Got_rsp[src] = True
    self.Childs.append(src)
    if self.got_rsp_from_all() and self.Parent is not None:
        self.send_msg("ACK", self.Parent)

def on_receive_nack(self, src: str):
    self.Got_rsp[src] = True

def got_rsp_from_all(self) -> bool:
    for src in self.Got_rsp:
        if not self.Got_rsp[src]:
            return False
    return True

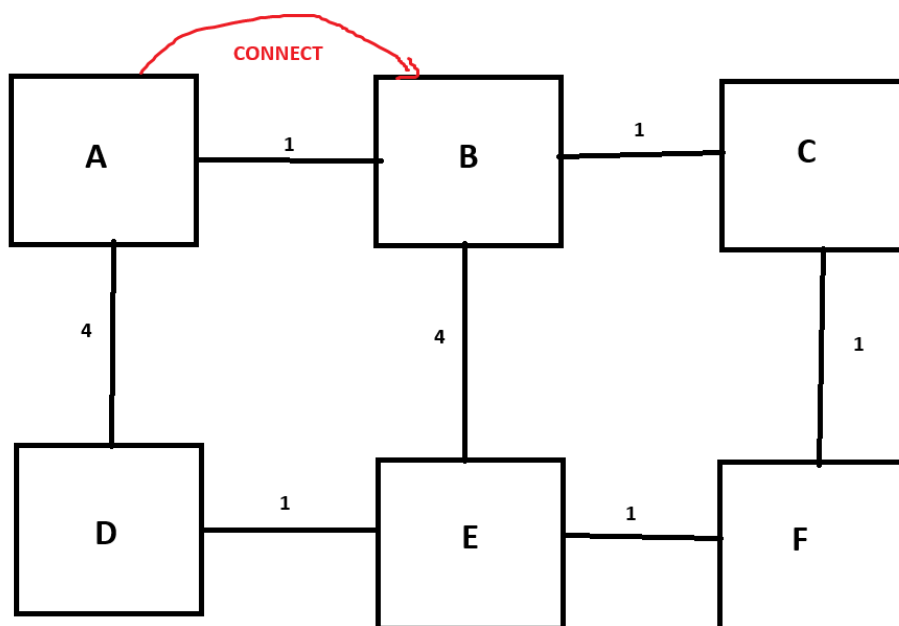
def terminate(self) -> bool:
    if self.got_rso_from_all:
        #DO TERMINATE
        return True

```

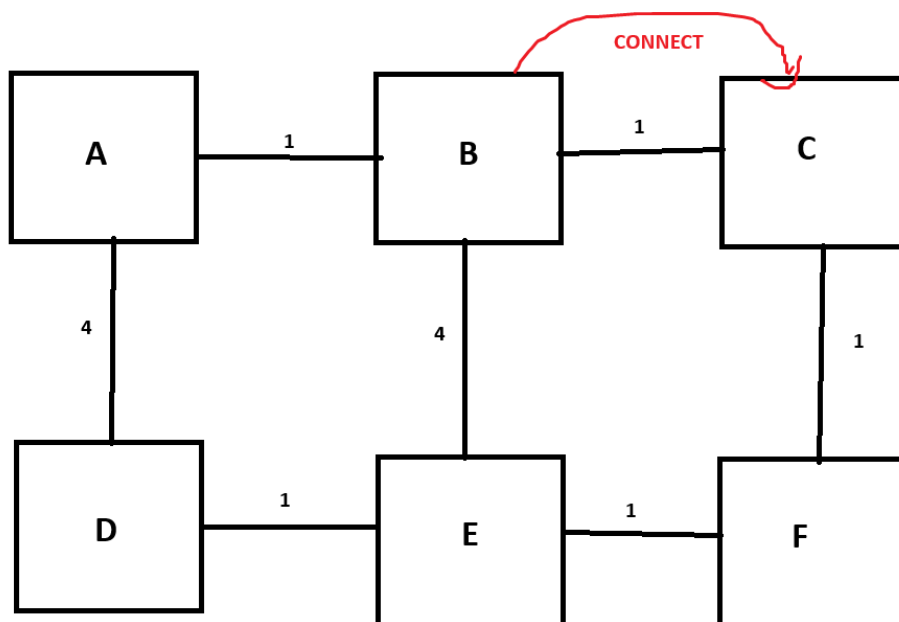
return False

סעיף ד':

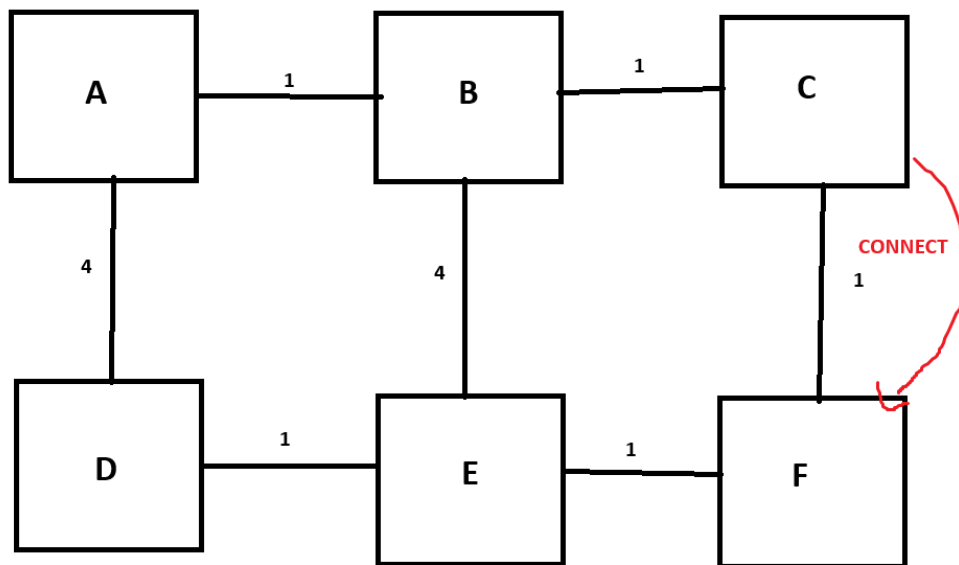
זמן 1:



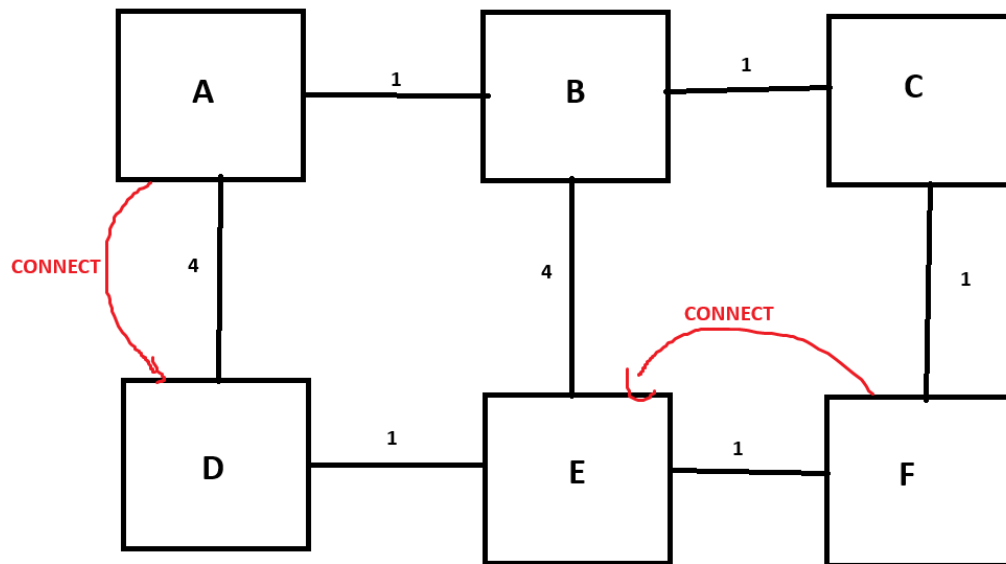
זמן 2:



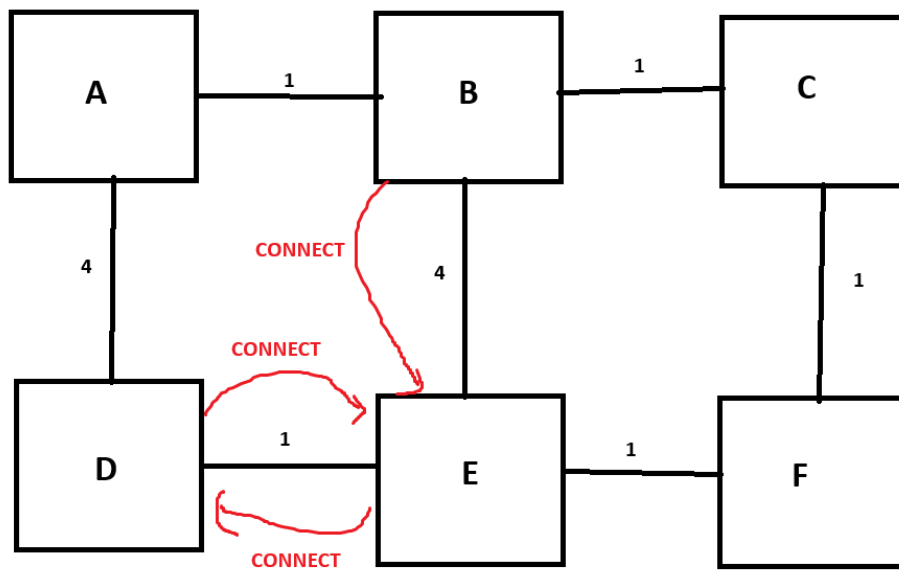
:3 μr



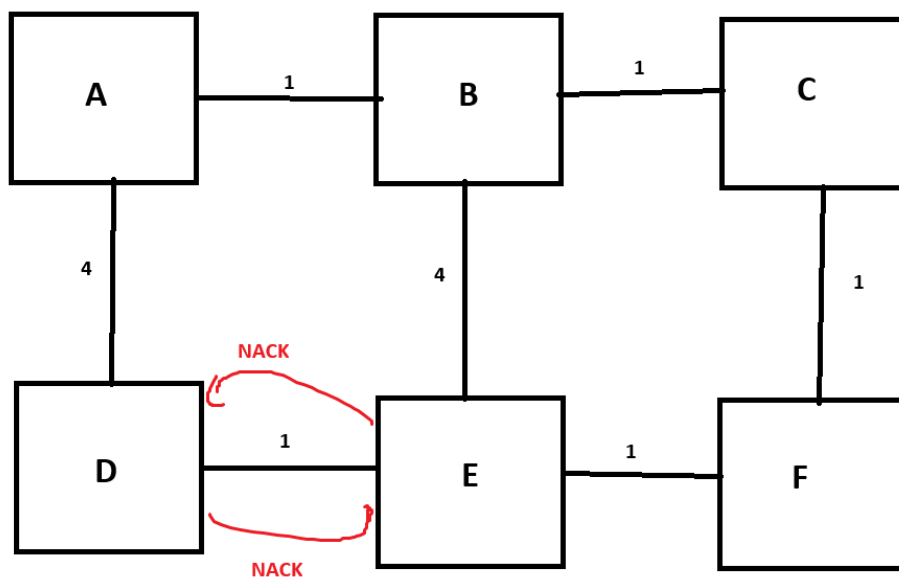
:4 μr



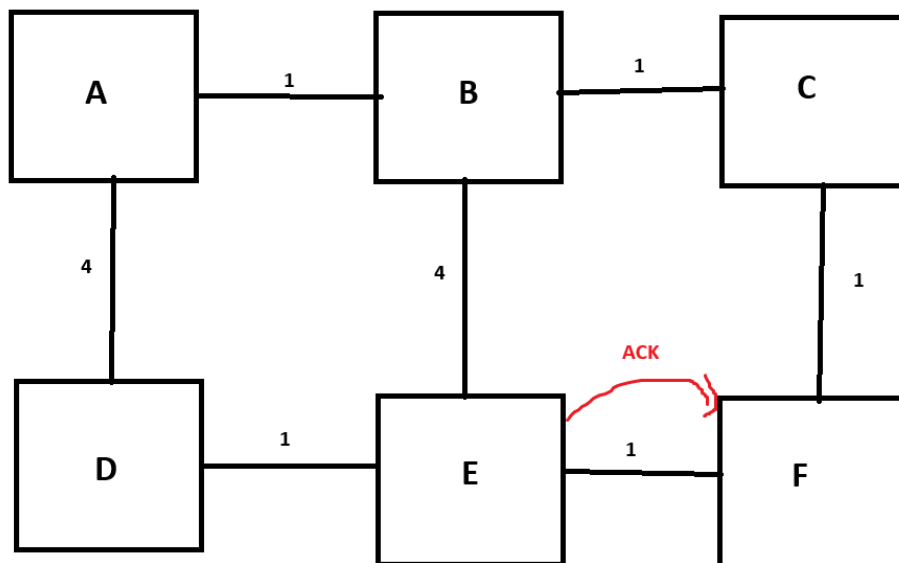
:5 μr



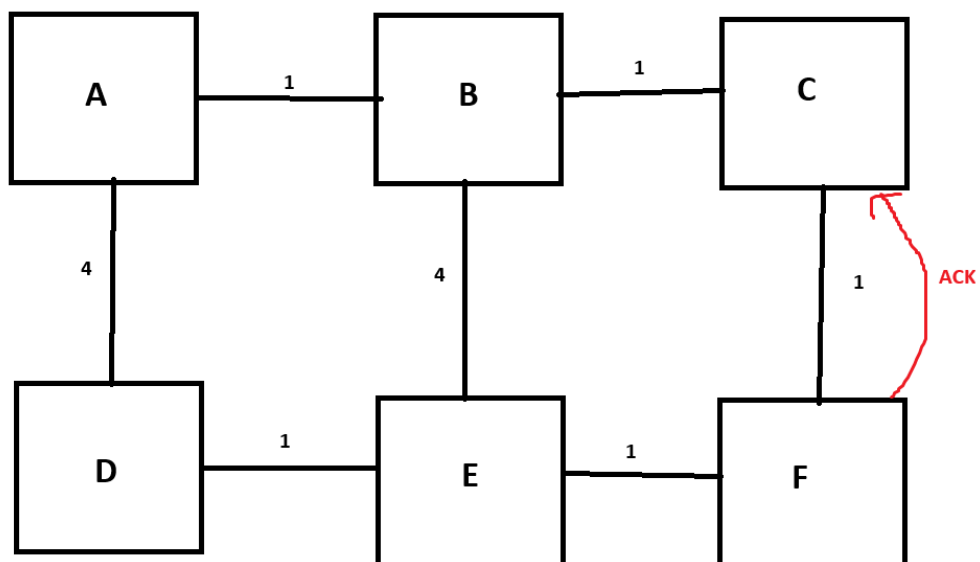
:6 μr



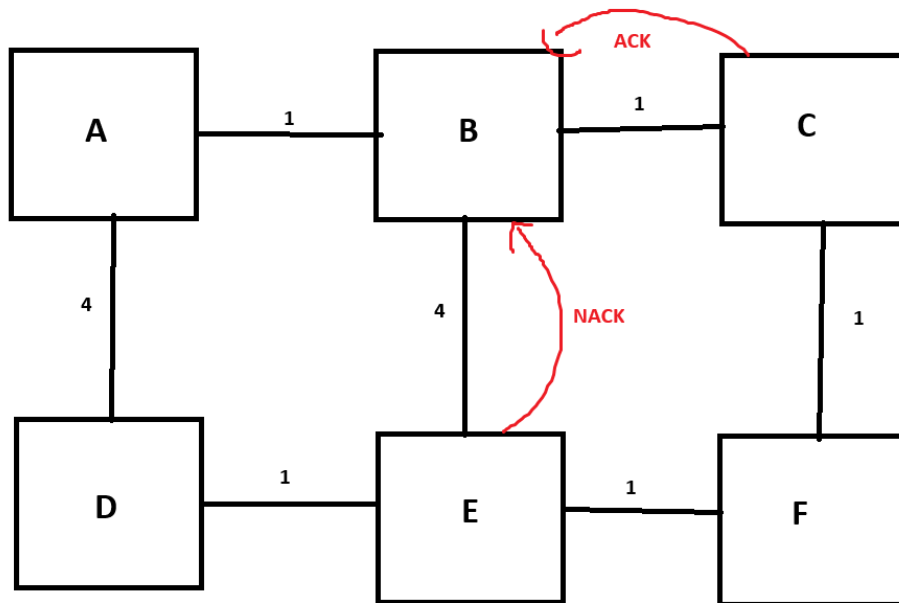
:7 μr



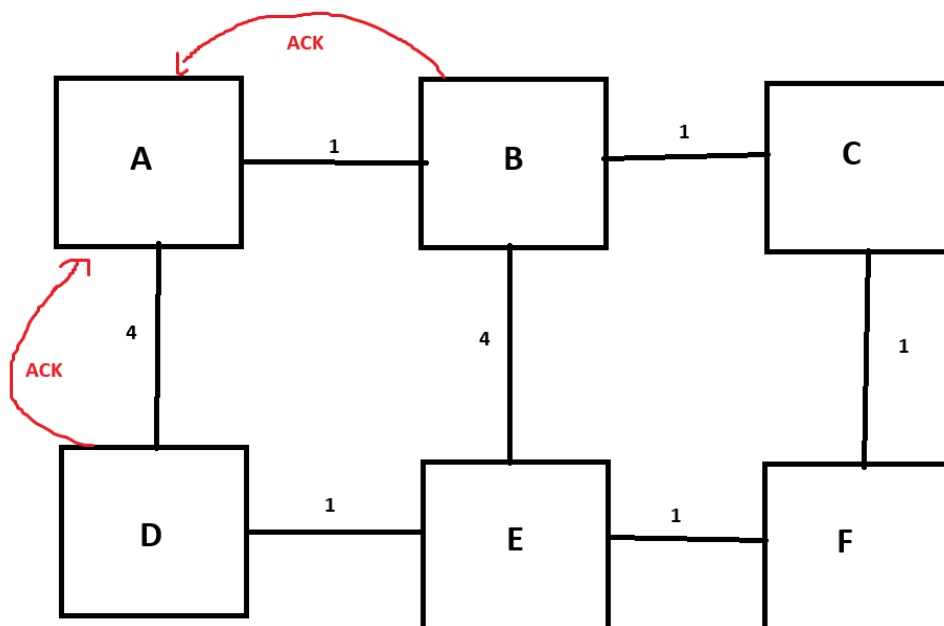
:8 μr



:9 *μs*



:10 *μs*



שאלה 2:

סעיף א:

```
class Router:
```

```
    def __init__(self, name: str, all_nodes: list[str], neighbors: dict):
```

```
        self.name = name
```

```
        self.all_nodes = all_nodes
```

```
        self.neighbors = neighbors # neighbors: dict[str, int] - maps neighbor name to cost
```

```
        self.routing_table = {}
```

```
        self.outbuf = {neighbor: [] for neighbor in neighbors}
```

```
        self.inbuf = {neighbor: [] for neighbor in neighbors}
```

```
        self.updated = False
```

```
        self.build_routing_table()
```

```
        self.broadcast_routing_table()
```

```
    def build_routing_table(self):
```

```
        all_nodes = self.all_nodes
```

```
        for dest in all_nodes:
```

```
            self.routing_table[dest] = {}
```

```
            for via in all_nodes:
```

```
                if dest == via:
```

```
                    if dest == self.name:
```

```
                        self.routing_table[dest][via] = 0
```

```
                    elif via in self.neighbors:
```

```
                        self.routing_table[dest][via] = self.neighbors[via]
```

```
                    else:
```

```
                        self.routing_table[dest][via] = float('inf')
```

```
            elif via == self.name:
```

```

        self.routing_table[dest][via] = None # can't route via self
    else:
        self.routing_table[dest][via] = float('inf')

def receive_update(self, msg: dict, src: str):
    for dest, cost in msg.items():
        new_cost = cost + self.neighbors[src] # Use actual link cost
        previous_cost = self.routing_table[dest][src]
        if new_cost != previous_cost:
            self.routing_table[dest][src] = new_cost
            self.updated = True

    if self.updated:
        self.broadcast_routing_table()
        self.updated = False

def best_routes(self) -> dict:
    best{} =
    for dest, via_map in self.routing_table.items():
        valid_routes = {via: cost for via, cost in via_map.items() if cost is not None and
cost < float('inf')}
        if valid_routes:
            best_via = min(valid_routes.items(), key=lambda item: item[1])
            best[dest] = best_via[1]
    return best

def broadcast_routing_table(self):
    best_routes = self.best_routes()
    for neighbor in self.neighbors:

```

```
self.outbuf[neighbor].append(best_routes)
```

```
def send_msg(self, dest: str, msg: dict):
```

```
#     Placeholder for actual message-passing mechanism  
    pass
```

```
def receive_from(self, src: str, msg: dict):
```

```
    self.inbuf[src].append(msg)  
    self.receive_update(msg, src)
```

סעיף ב':

ראוטר A יכול להתעדכן רק משכניו B ו D
במקרה שלנו A יעדיף לעבור דרך שכן B בשביל להגיע לקודקוד F – להלן שלבי התעדכנות של A
בטבלאות בכל מקום שיש תא רק הכוונה שיש שם ערך של אינסוף.

חישובים ש A עושה לפי הנוסחא הבא:

$$D(A,X) = \min \{ C(A,B) + D(B,X), C(A,D) + D(D,X) \}$$

$$D(A,B) = \min \{ C(A,B), D(A,D) + C(D,B) \}$$

$$D(A,D) = \min \{ C(A,D), D(A,B) + C(B,D) \}$$

זמן 1:

בטבלה הערכים של קודקוד A לאחר חישובים
העידכונים שכל צומת תשלח לאחר חישוב של כל אחד

From A	Via A	Via B	Via C	Via D	Via E	Via F
To A	NA	NA	NA	NA	NA	NA
To B	NA	1				
To C	NA					
To D	NA			4		
To E	NA					
To F	NA					

- צומת A –
 $D(A,B) = 1, D(A,D) = 4$ – כל השאר אינסוף
- צומת B –
 $D(B,A) = 1, D(B,C) = 1, D(B,E) = 4$ – כל השאר אינסוף
- צומת C –
 $D(C,B) = 1, D(C,F) = 1$ – כל השאר אינסוף
- צומת D –
 $D(D,A) = 4, D(D,E) = 1$ – כל השאר אינסוף
- צומת E –
 $D(E,D) = 1, D(E,B) = 4, D(E,F) = 1$ – כל השאר אינסוף
- צומת F –
 $D(F,C) = 1, D(F,E) = 1$ – כל השאר אינסוף

זמן 2:

בטבלה הערכים של קודקוד A לאחר חישובים
לאחר העדכונים להלן החישובים שצומת A תבצע לאחרת העידכונים הבאים

$$D(B,A) = 1, D(B,C) = 1, D(B,E) = 4 \text{ - צומת B}$$

$$D(D,A) = 4, D(D,E) = 1 \text{ - צומת D}$$

$$\begin{aligned} D(A,B) &= \min \{ 1, 4 + \text{INF} \} = 1 \quad \circ \\ D(A,C) &= \min \{ 1 + 1, 4 + \text{INF} \} = 2 \quad \circ \\ D(A,D) &= \min \{ 4, 1 + \text{INF} \} = 4 \quad \circ \\ D(A,E) &= \min \{ 1 + 4, 4 + 1 \} = 5 \quad \circ \end{aligned}$$

העידכונים שכל צומת תשלח לאחר חישוב של כל אחד:

From A	Via A	Via B	Via C	Via D	Via E	Via F
To A	NA	NA	NA	NA	NA	NA
To B	NA	1				
To C	NA	2				
To D	NA			4		
To E	NA	5		5		
To F	NA					

- צומת A –
 $D(A,B) = 1, D(A,C) = 2, D(A,D) = 4, D(A,E) = 5$ – כל השאר אינסוף
- צומת B –
 $D(B,A) = 1, D(B,C) = 1, D(B,E) = 4, D(B,F) = 2, D(B,D) = 5$ – כל השאר אינסוף
- צומת C –
 $D(C,B) = 1, D(C,F) = 1, D(C,A) = 2, D(C,E) = 2$ – כל השאר אינסוף
- צומת D –
 $D(D,A) = 4, D(D,E) = 1, D(D,B) = 5, D(D,F) = 2$ – כל השאר אינסוף

- צומת E - $D(E,D) = 1, D(E,B) = 4, D(E,F) = 1, D(E,A) = 5, D(E,C) = 2$
 - צומת F - $D(D,A) = 4, D(D,E) = 1, D(D,B) = 5, D(D,F) = 2$
- כל השאר אינסוף - $D(F,C) = 1, D(F,E) = 1, D(F,B) = 2, D(F,D) = 2$

זמן 3:

בטבלה הערכים של קודקוד A לאחר חישוב

לאחר העדכונים להלן החישובים שצומת A תבצע לאחרת העדכונים הבאים
 מצומת B - $D(B,A) = 1, D(B,C) = 1, D(B,E) = 4, D(B,F) = 2, D(B,D) = 5$
 מצומת D - $D(D,A) = 4, D(D,E) = 1, D(D,B) = 5, D(D,F) = 2$

$$\begin{aligned} D(A,B) &= \min \{ 1, 4 + 5 \} = 1 \\ D(A,C) &= \min \{ 1 + 1, 4 + \text{INF} \} = 2 \\ D(A,D) &= \min \{ 4, 1 + 5 \} = 4 \\ D(A,E) &= \min \{ 1 + 4, 4 + 1 \} = 5 \\ D(A,F) &= \min \{ 1 + 2, 4 + 2 \} = 3 \end{aligned}$$

העידכונים שכל צומת תשלח לאחר חישוב של כל אחד:

- צומת A - $D(A,B) = 1, D(A,C) = 2, D(A,D) = 4, D(A,E) = 5, D(A,F) = 3$
- צומת B - $D(B,A) = 1, D(B,C) = 1, D(B,E) = 3, D(B,F) = 2, D(B,D) = 5$
- צומת C - $D(C,B) = 1, D(C,F) = 1, D(C,A) = 2, D(C,D) = 3, D(C,E) = 2$
- צומת D - $D(D,A) = 4, D(D,E) = 1, D(D,B) = 5, D(D,C) = 3, D(D,F) = 2$
- צומת E - $D(E,D) = 1, D(E,B) = 3, D(E,F) = 1, D(E,A) = 5, D(E,C) = 2$
- צומת F - $D(F,C) = 1, D(F,E) = 1, D(F,B) = 2, D(F,D) = 2, D(F,A) = 3$

זמן 4:

בטבלה הערכים של קודקוד A לאחר חישוב

לאחר העדכונים להלן החישובים שצומת A תבצע לאחרת העדכונים הבאים
 מצומת B - $D(B,A) = 1, D(B,C) = 1, D(B,E) = 3, D(B,F) = 2, D(B,D) = 5$
 מצומת D - $D(D,A) = 4, D(D,E) = 1, D(D,B) = 5, D(D,C) = 3, D(D,F) = 2$

$$\begin{aligned} D(A,B) &= \min \{ 1, 4 + 5 \} = 1 \\ D(A,C) &= \min \{ 1 + 1, 4 + 3 \} = 2 \\ D(A,D) &= \min \{ 4, 1 + 5 \} = 4 \\ D(A,E) &= \min \{ 1 + 3, 4 + 1 \} = 4 \\ D(A,F) &= \min \{ 1 + 2, 4 + 2 \} = 3 \end{aligned}$$

העידכונים שכל צומת תשלח לאחר חישוב של כל אחד:

- צומת A - $D(A,B) = 1, D(A,C) = 2, D(A,D) = 4, D(A,E) = 5, D(A,F) = 3$
- צומת B - $D(B,A) = 1, D(B,C) = 1, D(B,E) = 3, D(B,F) = 2, D(B,D) = 4$
- צומת C - $D(C,B) = 1, D(C,F) = 1, D(C,A) = 2, D(C,D) = 3, D(C,E) = 2$
- צומת D - $D(D,A) = 4, D(D,E) = 1, D(D,B) = 4, D(D,C) = 3, D(D,F) = 2$

From A	Via A	Via B	Via C	Via D	Via E	Via F
To A	NA	NA	NA	NA	NA	NA
To B	NA	1		9		
To C	NA	2				
To D	NA	6		4		
To E	NA	5		5		
To F	NA	3		6		

From A	Via A	Via B	Via C	Via D	Via E	Via F
To A	NA	NA	NA	NA	NA	NA
To B	NA	1		9		
To C	NA	2		7		
To D	NA	6		4		
To E	NA	4		5		
To F	NA	3		6		

- צומת E - $D(E,D) = 1, D(E,B) = 3, D(E,F) = 1, D(E,A) = 4, D(E,C) = 2$
- צומת F - $D(F,C) = 1, D(F,E) = 1, D(F,B) = 2, D(F,D) = 2, D(F,A) = 3$

זמן 5:

From A	Via A	Via B	Via C	Via D	Via E	Via F
To A	NA	NA	NA	NA	NA	NA
To B	NA	1		8		
To C	NA	2		7		
To D	NA	5		4		
To E	NA	4		5		
To F	NA	3		6		

בטבלה הערכים של קודקוד A לאחר חישוב לאחר העדכונים להלן החישובים שצומת A תבצע לאחרת העדכונים הבאים
 מצומת B - $D(B,A) = 1, D(B,C) = 1, D(B,E) = 3, D(B,F) = 2, D(B,D) = 4$
 מצומת D - $D(D,A) = 4, D(D,E) = 1, D(D,B) = 4, D(D,C) = 3, D(D,F) = 2$

$$\begin{aligned} D(A,B) &= \min \{ 1, 4 + 4 \} = 1 \\ D(A,C) &= \min \{ 1 + 1, 4 + 3 \} = 2 \\ D(A,D) &= \min \{ 4, 1 + 4 \} = 4 \\ D(A,E) &= \min \{ 1 + 3, 4 + 1 \} = 4 \\ D(A,F) &= \min \{ 1 + 2, 4 + 2 \} = 3 \end{aligned}$$

העידכונים שכל צומת תשלח לאחר חישוב של כל אחד:

- צומת A - $D(A,B) = 1, D(A,C) = 2, D(A,D) = 4, D(A,E) = 5, D(A,F) = 3$
- צומת B - $D(B,A) = 1, D(B,C) = 1, D(B,E) = 3, D(B,F) = 2, D(B,D) = 4$
- צומת C - $D(C,B) = 1, D(C,F) = 1, D(C,A) = 2, D(C,D) = 3, D(C,E) = 2$
- צומת D - $D(D,A) = 4, D(D,E) = 1, D(D,B) = 4, D(D,C) = 3, D(D,F) = 2$
- צומת E - $D(E,D) = 1, D(E,B) = 3, D(E,F) = 1, D(E,A) = 4, D(E,C) = 2$
- צומת F - $D(F,C) = 1, D(F,E) = 1, D(F,B) = 2, D(F,D) = 2, D(F,A) = 3$

לאחר זמן זה יתבעו שוב ושוב עידכונים אבל שום דבר לא ישתנה אלא אם כן יקרה משהו במערכת.
 לכן ניתן לומר ש A יעביר מידע ל F דרך קודקוד B מאחר והעלות של מעבר ל F דרך B נמוך יותר.

סעיף ג':

ברגע השינוי וברגע שקודקודים B ו C יתוודעו לשינוי הן יבצעו עידכון למשקולות שלהן שיחלחלו ברשת כמו שראינו בסעיף הקודם, רק שעלול להיווצר לופים מאוד ארוכים עד שהערכים יתעדכו עם הערכים הנכונים מאחר שלדוגמא, כשקודקוד C ירצה לעדכן את העלות מסלול שלו ל B עם העלות החדשה שנוצרה הוא יבצע את הנוסחא הבא:

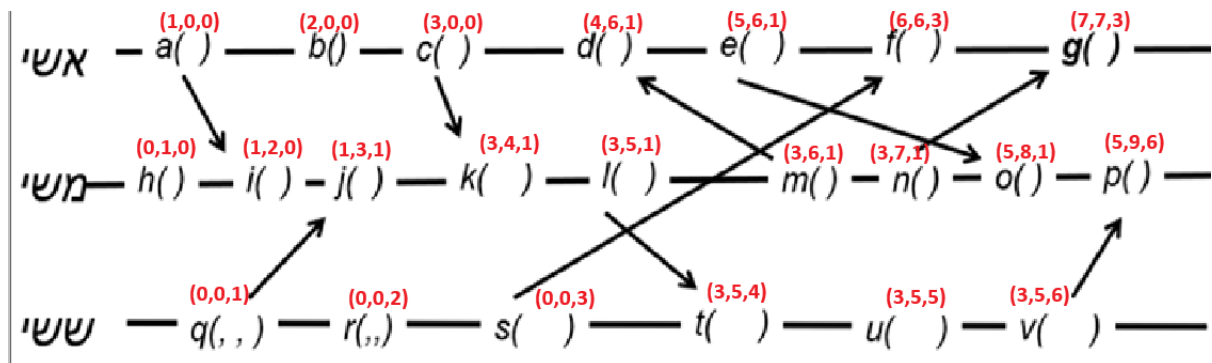
$$D(C,B) = \min \{ D(C,B), D(C,F) + D(F,B) \} = \{ 15, 1 + 2 \} = 3$$

שזו כמובן טעות, בעצם C לא יודע ש F עובר ל B דרך C עצמו ובעצם גם הוא מושפע מהשינוי הזה. ואז יוצר לופ כי C יעדכן את F שהעלות החדשה היא 3 ואז F יעדכן את העלות החדשה שלו ל 4 וכך זה יחזור עד שיגיע לערכים הנכונים.

הפתרון הוא שכל צומת שמפרסת את העלויות שלה תעביר ערך אינסוף לשכנים שלה, במקומות שבהם הדרך שלהם אל היעד עוברת דרך השכן ששלחתי לו.
 במקרה שלנו F לא יגיד ל C שהעלות שלו ל B היא 2 אלא יפרסם ל C שהעלות ל B היא אינסוף מאחר ו F עובר דרך C כדי להגיע ל B, אך ל E הוא יפרסם שהעלות היא 2 מאחר ו F מגיע ל B לא דרך E.

זזה יפתור את הבעיה מאחר ובמקרה שלנו הנוסחא תהיה כעת:
 $D(C,B) = \min \{ D(C,B) , D(C,F) + D(F,B) \} = \{15, 1 + \text{INF}\} = 15$
 שזה בדיוק מה שהיה צריך להתבצע ובכך חסכנו לופים רבים.

שאלה 3:
 סעיף א:



סעיף ב':

ניתן לראות של אשי חסרים 2 אירועים של משי ו 3 אירועים של ששי.
 למשי חסרים 2 אירועים של אשי ולא חסר שום אירוע של ששי.
 לששי חסרים 4 אירועים של אשי ו 4 אירועים של משי.
 לכן ניתן להגיד, שאין אפליקציה שמכירה את כל האירועים, אך ניתן להגיד שמשי מכירה הכי הרבה אירועים לאחר מכן אשי ולאחר מכן ששי.

סעיף ג':

טבלה של אשי

Event Id	Description	Vector Clock Timestamp
a_ashi	Send msg AA to Mashi	[1,0,0]
b_ashi	Compute "BBB"	[2,0,0]
c_ashi	Send msg CC to Mashi	[3,0,0]
d_ashi	Received msg "MM" from Mashi	[4,6,1]
e_ashi	Send msg EE to Mashi	[5,6,1]
f_ashi	Received msg "SS" from Sasi	[6,6,3]
g_ashi	Received msg "NN" from Sasi	[7,7,3]

טבלה של משי

Event Id	Description	Vector Clock Timestamp
h_mashi	Compute "Start mashi"	[0,1,0]
i_mashi	Received Msg AA from Ashi	[1,2,0]
j_mashi	Received Msg QQ from Sasi	[1,3,1]
k_mashi	Received Msg CC from Ashi	[3,4,1]
l_mashi	Send msg LL to Sasi	[3,5,1]
m_mashi	Send msg MM to Ashi	[3,6,1]
n_mashi	Send msg NN to Ashi	[3,7,1]
o_mashi	Received Msg EE from Ashi	[5,8,1]
p_mashi	Received Msg VV from Sasi	[5,9,6]

טבלה של ששי

Event Id	Description	Vector Clock Timestamp
q_sasi	Send msg QQ to Mashi	[0,0,1]
r_sasi	Compute "RR"	[0,0,2]
s_sasi	Send msg SS to Ashi	[0,0,3]
t_sasi	Received Msg LL from Mashi	[3,5,4]
u_sasi	Compute "UU"	[3,5,5]
v_sasi	Send msg VV to Mashi	[3,5,6]

סעיף ד':

להלן הקובץ המאוחד מסודר לפי ה causal order – כל האירועים המקביליים היו יכולים להיות מסודרים אחרת, סידרתי אותם בצורה מסויימת אך זה גם היה יכול להיות אחרת.

Event Id	Description	Vector Clock Timestamp	Concurrent to
a_ashi	Send msg AA to Mashi	[1, 0, 0]	H,Q,R,S
h_mashi	Compute "Start mashi"	[0, 1, 0]	A,B,C,Q,R,S
q_sasi	Send msg QQ to Mashi	[0,0,1]	A,B,C,H,I
b_ashi	Compute "BBB"	[2, 0, 0]	Q,R,S,H,I,J
i_mashi	Received Msg AA from Ashi	[1, 2, 0]	A,B,C,Q,R,S
r_sasi	Compute "RR"	[0, 0, 2]	A, B, C, D, E, H, I, J, K, L, M, N, O
c_ashi	Send msg CC to Mashi	[3, 0, 0]	Q,R,S,H,I,J
j_mashi	Received Msg QQ from Sasi	[1, 3, 1]	A,B,C,Q,R,S
s_sasi	Send msg SS to Ashi	[0, 0, 3]	A, B, C, D, E, H, I, J, K, L, M, N, O
k_mashi	Received Msg CC from Ashi	[3, 4, 1]	R,S
l_mashi	Send msg LL to Sasi	[3, 5, 1]	R,S

m_mashi	Send msg MM to Ashi	[3, 6, 1]	R,S,T,U,V
t_sasi	Received Msg LL from Mashi	[3, 5, 4]	D,E,F,G,M,N,O
d_ashi	Received msg "MM" from Mashi	[4, 6, 1]	N,R,S,T,U,V
n_mashi	Send msg NN to Ashi	[3, 7, 1]	D,E,F,R,S,T,U,V
u_sasi	Compute "UU"	[3, 5, 5]	D,E,F,G,M,N,O
e_ashi	Send msg EE to Mashi	[5, 6, 1]	N,R,S,T,U,V
v_sasi	Send msg VV to Mashi	[3, 5, 6]	D,E,F,G,M,N,O
f_ashi	Received msg "SS" from Sasi	[6, 6, 3]	N, O,P,T,U,V
o_mashi	Received Msg EE from Ashi	[5, 8, 1]	F,G,R,S,T,U,V
g_ashi	Received msg "NN" from Sasi	[7, 7, 3]	O,P,T,U,V
p_mashi	Received Msg VV from Sasi	[5, 9, 6]	F,G