# תרגיל בית 3

ת.ז.:

211709597 212699581

# שאלה 1:

-1

Algorithm (Shortest path not in u):

Input: A directed graph G = (V, E) and two vertices s, u ∈ $V$

Algorithm:

Build G' = (V', E') from G = (V, E) such that V' = { $v \in V \mid v \neq u$ }

and E' = { $(v', u') \in E \mid v' \neq u \; and \; u' \neq u$ }

Run BFS(G', s)

Time complexity:

Constructing G' from G - O(2V + E)

construct V' by scanning V - O(V)

and E by scanning the adjacency list - O(V + E)

Run BFS - O(V + E)

In total, the run time is O(2V + 2E) = O(V + E)

Correctness:

The goal: show that when the algorithm terminates:

$$v.d = \delta(s, v) \; that \; doesn't \; go \; through \; u, \; for \; every \; vertex \; v \in V$$

from the BFS algorithm it initiate every v.d with ∞ and if v is not reachable from v it's d field stays ∞

and from the correctness of BFS we get that BFS on G' and s (that doesn't have u) that $v.d = \delta(s, v)$ and that path doesn't include u since it's not a vertex in the graph.

Input: A directed graph G = (V, E) and three vertices $s,\ u_1,\ u_2 \in V$

Algorithm:

There's four possible ways to get a limited path

if the path doesn't contain u1 and u2

if the path contains only u1 (or only u2)

if the path contains u1 and u2 and the first appearance of u1 is before the first appearance of u2

1- Let $G_{u1},\ G_{u2}$

$G_{u1} = (V_{u1},\ E_{u1})\ such\ that\ G_{u1}$ = shortest_path_not_in_u(G, s, u1)

$G_{u2} = (V_{u2}, E_{u2})\ such\ that\ G_{u2}$ = shortest_path_not_in_u(G, s, u2)

2- construct $G'_{u12} = (V'_{u12},\ E'_{u12})$ as follows $V'_{u12} = \{v \in V :\ v \neq u_1\}$

$E'_{u12} = \{(v,\ u) \in E :\ v \neq u_1 \wedge u \neq u_1\}$

3- Let $G_{u12}$ = shortest_path_not_in_u($G'_{u12},\ s,\ u2$)

4- Construct $G_0$ from G by taking $V_0 = \{v \in V \mid v \neq u_2\}$

$E_0 = \{(v,\ u) \in E \mid v \neq u_2 \wedge u \neq u_2\}$

5- Run BFS on $G_0$ and s

if $u_1.d = \infty$

P' = NIL

else

6- Construct G' tree from the $G_0$.Transpose

7- Find the path from s to $u_1$ denote by P' (if v.d = ∞ we return NIL)

8- Construct $G_1$ graph from G such that $V_1 = \{v \in V \mid v \notin P' \setminus \{u_1\}\}$

$E_1 = \{(v,\ u) \in E' \mid v \in V_1 \wedge\ u \in V_1\}$

9- Run BFS on $G_1$ and $u_1$

if P’ = NIL or $u_2.d = \infty$

P’’ = NIL

else

10- Construct G’’ tree from $G_1$.Transpose

11- Find the path from $u_1$ $to$ $u_2$ denote by P’’

12- Construct G’’’ graph from G’’ such that $V''' = \{v \in V \mid v \notin P''\}$

$E''' = \{(v, u) \in E'' \mid v \in V''' \wedge u \in V'''\}$

13- Run BFS on G’’’ and $u_2$

14- Construct the graph A as follows

$V_A = P' \cup P'' \cup V'''$

$E_A = \{(v, u) \in E''' \cup P'' \cup P' \mid v \in P' \cup P'' \cup V''' \wedge u \in P' \cup P'' \cup V'''\}$

15- go through the adjacency list and update v.d as follows:

if P’’ = NIL

$v.d = min\{v_{G_{u12}}.d, v_{G_{u1}}.d, v_{G_{u2}}.d\}$

else

$v.d = min\{v_A.d, v_{G_{u12}}.d, v_{G_{u1}}.d, v_{G_{u2}}.d\}$

Time complexity:

(Note that $E'' \leq E' \leq E$ $and$ $E_A \leq E$)

1- 2 * O(V + E)

2-6 O(V + E)

7- O(E)

8-10 O(V + E)

11- O(E)

12-15 O(V + E)

so we got O(12V + 14E) = O(V + E)

Correctness:

The goal: show that when the algorithm terminate

$$v.d = \delta(s, v) \ that\ doesn't\ have\ u_2\ before\ u_1\ in\ the\ path\ from\ s\ to\ v$$

$$for\ every\ vertex\ v \in V$$

there's four cases:

from the correctness of the shortest_path_not_in_u we get

$G_{u1}$, $G_{u2}$ with $v.d = \delta(s, v)$ which don't include u1, u2 respectively, w.l.o.g , for$G_{u1}$, if every path from s to v, $\forall\, v \in V$, have u1, from the correctness of BFS (algorithm in 1.1) v.d will be ∞

$G_{u12}$: since we're getting this tree from $G'_{u12}$ that doesn't have u1 and the running the shortest_path_not_in_u algorithm with $G'_{u12}$ and s, u2 and from the correctness of this algorithm we get the shortest path that doesn't include u1, u2, $\forall\, v \in V$

$G_A$: from the correctness of BFS and Algo 1.1, if there is a path that doesn't include $u_2$ from s to v, $\forall\, v \in V$, the algorithm will find this path, because in line 4 we construct $G_0$ by removing $u_2$ from G, then running BFS on $G_0$ with s, if there's not the P' will be NIL => P'' will be NIL, and if there's no path between $u_1$and $u_2$P'' will be NIL

in line 15 we check if P'' = NIL, if it is NIL that means there's no path between s and v, $\forall\, v \in V$, that have the first appearance of $u_1$before the first appearance of $u_2$which means v.d should be ∞

thus calculating v.d inside the if in line 15 is equivalent to
$v.d = min\{v_A.d,\ v_{G_{u12}}.d,\ v_{G_{u1}}.d,\ v_{G_{u2}}.d\}$where $v_A.d$=∞

# שאלה 2:

**הכיוון הראשון:**

לפי משפט $u.d < v.d < v.f < u.f$ אם"ם u אב קדמון של v

ולכן u אב קדמון ל w ו w אב קדמון ל v

וממשפט אם קיים מסלול בין u ל v אזי קיים מסלול פשוט בין u ל v

ולכן בין u ו w קיים מסלול פשוט P

ולכן בין w ו v קיים מסלול פשוט K

ולכן $< \mathrm{P}, \mathrm{K} >$ מסלול מ $u$ ל $v$ שעובר ב $w$, ולכן לפי אותו משפט קיים מסלול פשוט מ $u$ ל $v$ שעובר ב $w$

**הכיוון השני:**

נתון שיש מסלול פשוט מ $u$ ל $v$ שעובר ב $w$ ולכן u אב קדמון של v ו w ,

ו w אב קדמון של v .

ולכן קיימת הרצת DFS מ u כך שלפי משפט הסוגריים מתקיים: $u.d < v.d < v.f < u.f$ .

## שאלה 3:

(1) מכיוון שהגרף קשיר אז לכל הרצת *DFS* קיים מסלול בין *u* ל *v*.
בנוסף *DFS* עובר על כל הצמתים לפני שמתחיל לחזור עליהם (*CONTRACT*)
ולכן אם $w$ בן ל $s$ אז $w.d = s.d + 1$ .
כלומר $u.d = v.d + d(u,v)$
מהגדרת המרחק הקצר ביותר: מהגדרת המרחק הקצר ביותר:
$\delta(u,v) \leq d(u,v)$
ולכן נקבל: $d(u,v) = u.d - v.d \geq \delta(u,v)$ .

(2) נוכיח שלכל $u,v,w \in V$ כך ש $u.d > v.d$ מתקיים: $\delta(u,v) \geq |\delta(u,w) - \delta(v,w)|$.

נחלק למקרים:

1-אם $u.d > v.d > w.d$ אזי מסעיף הקודם נקבל:

$$\delta(u,v) = |\delta(u,v)| = |\delta(u,w) - \delta(v,w)|$$

2- אם $u.d > w.d > v.d$ אזי:

$$\delta(u,v) = \delta(u,w) + \delta(v,w) = \delta(v,w) + \delta(u,w) > \delta(u,w) > \delta(u,w) - \delta(v,w)$$

3-אם $w.d>u.d>v.d$ אזי:

$$\delta(u,v) = |\delta(u,v)| = |\delta(v,w) - \delta(u,w)| = |\delta(u,w) - \delta(v,w)|$$

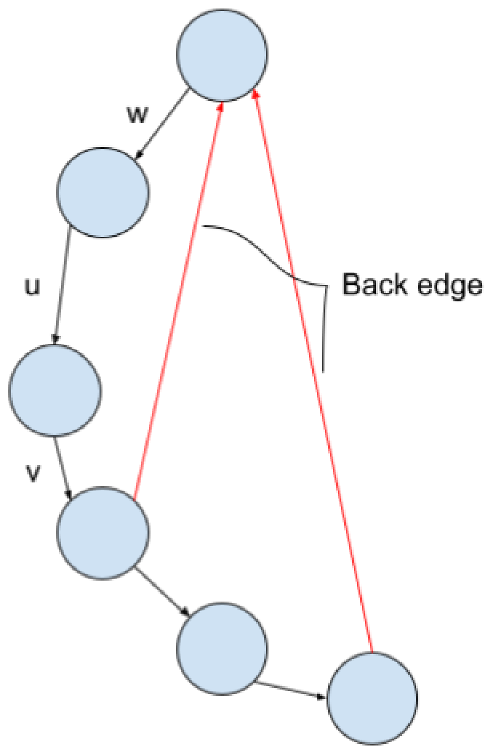
עכשיו נוכיח המבוקש:

$$\alpha(u.d - v.d) > u.d - v.d \geq |\delta(u,w) - \delta(v,w)| \geq \delta(u,w) - \delta(v,w)$$

המעבר הראשון מתקבל כיוון $\alpha > 1$.

## שאלה 4:

הפרכה -דוגמה נגדית :




אם נמחק אחד משלושת הקשתות u, w, v נקבל גרף חסר מעגלים כלומר הטענה אינה נכונה כי גודל
קשת אחת הוא 1 < מגודל |F|=2

## שאלה 5:

1-יהי $x, y \epsilon V$ כך ש: $\delta(x,y) = r(G)$ .

ויהיו $u, v \epsilon V$ כך ש$\delta(u,v) = D(G)$.

מהגדרת הקוטר נקבל מייד: $r(G) \le D(G)$

$r$ אז לכל $x \epsilon V$ מתקיים:

$\varepsilon(x) \ge \delta(x,u)$(1)

$\varepsilon(x) \ge \delta(x,v)$(2)

אזי: $\forall\ x \in V:\ D(G) = \delta(u,v) = \delta(u,x) + \delta(x,v) \le 2 * \varepsilon(x)$

ובפרט עבור : $\min\{\varepsilon(x)\} = r(G)$

קיבלנו ש : $D(G) \le 2 * r(G) \rightarrow \frac{1}{2} * D(G) \le r(G)$

בסה"כ קיבלנו $D(G) \leq r(G) \leq D(G) * \frac{1}{2}$.

2- **הוכחה:**

**הכיוון הראשון:** יהי $v \epsilon V$ מרכז העץ $T$ .

מסעיף(1) נקבל שקיים צומת $u$ שעבורו $\delta(u,v) = \frac{1}{2} * \lceil D(T) \rceil$ שבעצם הוא הצומת הכי רחוק מ $v$ כלומר $\varepsilon(v) = \delta(u,v) = \frac{1}{2} * \lceil D(T) \rceil$

וקיים צומת $w$ שעבורו: $\delta(u,w) = \varepsilon(u) = D(T)$ (כלומר בינו לבין $u$ מתקבל קוטר הגרף)

נניח בשלילה ש $v$ אינו נמצא במסלול $< u,w >$ ונסמן ב $x$ הצומת שבמסלול הקרוב ביותר על $v$ .

מתקיים :

$$\delta(u,x) < \frac{1}{2} * \lceil D(T) \rceil$$

ולכן : $\delta(x,w) \geq \frac{1}{2} * \lceil D(T) \rceil$

$$D(T) = \delta(u,w) = \delta(u,v) + \delta(v,x) + \delta(x,w) > \delta(u,v) + \delta(x,w) \geq \frac{1}{2} * \lceil D(T) \rceil + \frac{1}{2} * \lceil D(T) \rceil \geq D(T)$$

**בסתירה ,ולכן ההנחה שגוייה כלומר** ש $v$ נמצא במסלול $< u,w >$.

**הכיוון השני:** נניח בשלילה שקיים צומת $y$ שעבורו יש מסלול ארוך יותר עם $v$ מאשר עם $u$ כלומר: $\varepsilon(v) = \delta(y,v) > \frac{1}{2} * \lceil D(T) \rceil$

במקרה ש $u$ נמצא על המסלול $< v,y >$ אז:

$$D(T) = \delta(u,v) > \delta(y,v) = \delta(y,w) + \delta(w,v) > \frac{1}{2} * \lfloor D(T) \rfloor + \frac{1}{2} * \lceil D(T) \rceil = D(T)$$

ובמקרה ש $u$ לא נמצא על המסלול $< v,y >$ אז:

$$\delta(y,u) = \delta(y,v) + \delta(v,u) > \frac{1}{2} * \lceil D(T) \rceil + \frac{1}{2} * \lceil D(T) \rceil \geq D(T)$$

בשני מקרים המשלימים זה לזה נקבל סתירה כי אורך המסלולים גדולים מאורך הקוטר בסתירה להגדרת הקוטר.

**3-** האלגוריתם:

נבחר צומת $s \epsilon V$ שרירותי כלשהו

$BFS(T,s)$-1

$\forall x \epsilon V$: $u$=$arg \{max\, \delta(s,x)\}$-2

$BFS(T,u)$-3

$\forall x \epsilon V$: $w$=$arg \{max\, \delta(u,x)\}$ -4

-5 נחפש הצומת $v \epsilon V$שבמסלול $< u, v >$ כך ש:

$\delta(v,u) = \frac{1}{2} * \lceil D(T) \rceil$ וגם

$\delta(v,w) = \frac{1}{2} * \lfloor (D(T) \rfloor$

ניתוח זמן :

כל ריצה של $BFS$ לוקחת $O(|V|+|E|)$ומכיוון שהגרף הנתון הוא גרף אז מתקיים :

$|E|=|V|-1=n-1$

ולכן השורות 1+3 לוקחות $O(n)$

כדי למצוא המרחק המקסימלי מבין כל המרחקים צריך לעבור על כל הצמתים ולכן שורות 2+4 לוקחות $O(n)$

אחרי שמצאנו הצמתים $u,v$ נחפש במסלול שביניהם שהמרחק שלו לכל היותר $|E|$ ולפחות $|V|$ ולכך שורה 5 לוקחת גם $O(n)$

בסה"כ האלגוריתם לוקח O(n) יחידות זמן.

נכונות האלגוריתם:

לפי הכיוון השני ממשפט בסעיף 5-2 האלגוריתם מחזיר צומת v ∈ V המהווה מרכז שלT

. בהינתן עץ $T = (V, E)$, צומת $v \in V$ הוא מרכז של $T$ אם ורק אם קיימים שני צמתים $u, w \in V$ כך ש־$\delta(u,w) = D(T)$ ומתקיים ש־$v$ נמצא על המסלול הפשוט הייחודי בין $u$ ל־$w$ כך ש־$\delta(u,v) = \lceil \frac{D(T)}{2} \rceil$, $\delta(v,w) = \lfloor \frac{D(T)}{2} \rfloor$