

ברוכים הבאים לתרגול 8 😊

שחר אנגל

shaharbel0@gmail.com

תרגול- ימי שני 14-16 וימי חמישי 13-15



נושא התרגול

Breadth-First Search -BFS ■



■ לפני שנתחיל לדבר על האלגוריתם, ניזכר איך אפשר לממש גרף בתכנות:

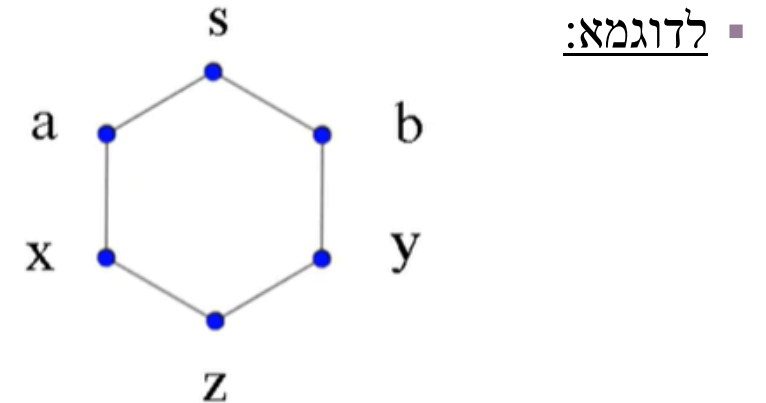
1. להשתמש באובייקטים ולייצר מחלקות: Edge, Vertex, Graph וכו'

2. להשתמש במטריצת שכנויות

3. להשתמש ברשימת שכנויות



- הרעיון שעומד מאחורי האלגוריתם הוא סריקה לפי שכבות:
- בהינתן גרף $G=(V,E)$ וקודקוד מסוים s המשמש כמקור, חיפוש לרוחב בוחן בשיטתיות את הקשתות ב- G כדי "לגלות" כל קודקוד שניתן להגיע אליו מ- s .
- הוא מחשב את המרחק (מספר הקשתות המינימלי) מ- s לכל הקודקודים שניתן להגיע אליהם מ- s .
- האלגוריתם פועל גם על גרפים מכוונים וגם על גרפים לא מכוונים.



- בהינתן הגרף הזה ניתן לסרוק אותו ב-2 דרכים:

■ s, a, b, x, y, z

■ s, b, a, y, x, z



- האלגוריתם מחלק את הקודקודים ל-3 צבעים:

- לבן- הקודקוד טרם התגלה

- אפור- הקודקוד התגלה ולא טופל

- שחור- הקודקוד טופל

- הוא בונה עץ רוחב ששורשו הוא קודקוד המקור s ולפי השכבות והמרחקים ממנו הוא בונה את שאר הגרף. לכן, קודקוד במרחק k יתגלה לפני קודקוד במרחק $k+1$.

- כל קודקוד שמגלה את שכניו הופך להיות האבא שלהם. בגלל שכל קודקוד מתגלה פעם אחת אז יש לו רק אבא אחד.

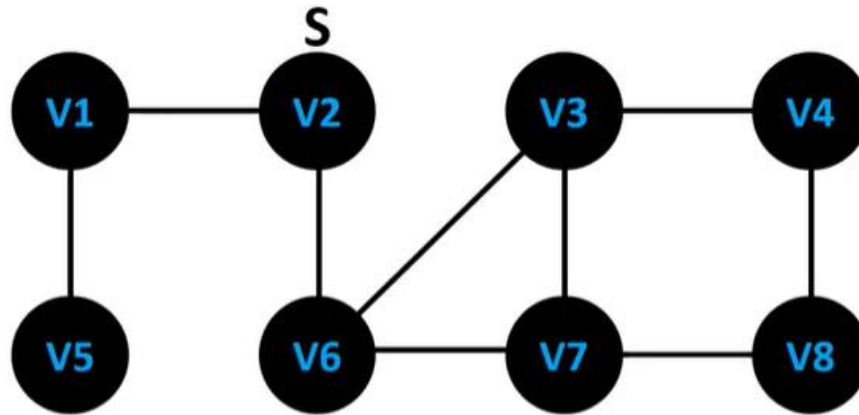
- המסלול שיוצא מ- s לקודקוד v כלשהו בעץ רוחב, מקביל למסלול הקצר ביותר בין s ל- v בגרף.



■ פסאודו קוד: (נניח שהגרף מיוצג לפי רשימת שכנויות)

- BFS(G,s):
- for each vertex in $V(G) \setminus \{s\}$
 - $\text{color}[u] = \text{WHITE}$
 - $d[u] = \infty$
 - $f[u] = \text{null}$
- $\text{color}[s] = \text{GRAY}$
- $d[s] = 0$
- $f[s] = \text{null}$
- Queue q
- Enqueue(q, s)
- while q not empty:
 - $u = \text{dequeue}(q)$
 - for each v in $N(u)$:
 - if $\text{color}[v] = \text{WHITE}$:
 - $\text{color}[v] = \text{GRAY}$
 - $d[v] = d[u] + 1$
 - $f[v] = u$
 - Enqueue(q, v)
 - $\text{color}[u] = \text{BLACK}$





color =



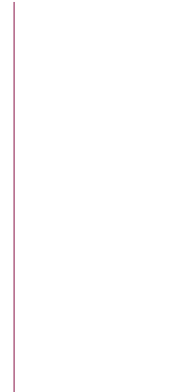
d =



f =



Queue



■ פסאודו קוד: (נניח שהגרף מיוצג לפי רשימת שכנויות)

■ BFS(G,s):

$o(|V|)$ {

- for each vertex in $V(G) \setminus \{s\}$
 - $color[u] = WHITE$
 - $d[u] = \infty$
 - $f[u] = null$

■ $color[s] = GRAY$

■ $d[s] = 0$

■ $f[s] = null$

■ Queue q

■ Enqueue(q, s)

■ while q not empty:

$o(|V|)$

■ $u = dequeue(q)$

■ for each v in N(u):

■ if $color[v] = WHITE$:

■ $d[v] = d[u] + 1$

■ $f[v] = u$

■ Enqueue(q, v)

■ $color[u] = BLACK$

$o(|E|)$

$o(deg(v))$

סיבוכיות:

- אנו צובעים כל קודקוד רק פעם אחת בלבן ולכן מובטח לנו שכל קודקוד יכנס פעם אחת לתור ויצא רק פעם אחת מהתור - הוצאה והכנסה $= o(1)$, לכן נקבל

$o(|V|)$

- כל איטרציה, אנו עוברים על השכנים של אותו הקודקוד ולכן נעבור על

$o(deg(v))$

- או במילים אחרות נקבל: $o(|E|)$

- סה"כ נקבל: $o(|V+E|)$



- מסקנות:
- לאחר הרצת האלגוריתם $BFS(G,s)$ בגרף $G=(V,E)$ נאמר ש:
- כל רכיב הקשירות של s התגלה
- מערך d מכיל את המסלולים הקצרים ביותר מ- s לכל קודקוד
- דרך מערך f ניתן לשחזר את המסלול הקצר ביותר

▪ כעת, איך ניצור או נדפיס מסלול?

▪ בעזרת אלגוריתם רקורסיבי:

- Print-Path(G,s,v):
- if $v=s$
 - print s
- else if $f[v] = \text{null}$:
 - print “no path from” s “to” v “exists”
- else
 - Print-Path($G,s,f[v]$)
- print v

סיבוכיות:
 $O(|V|)$



■ שאלות:

1. איך אפשר לדעת האם הגרף קשיר?

■ אם לאחר הרצת האלגוריתם כל הקודקודים צבועים בשחור

■ או אם יש רק null אחד במערך f

■ או אם כל המספרים במערך d סופיים

2. איך אפשר לדעת כמה רכיבי קשירות יש?

■ אם לאחר הרצת האלגוריתם יש עוד קודקודים לבנים, נפעיל את האלגוריתם מחדש על אחד הלבנים ונספור שזו איטרציה שניה.

נמשיך כך עד שכל הקודקודים נצבעים בשחור

3. איך אפשר לדעת מיהם הקודקודים בכל רכיב קשירות?

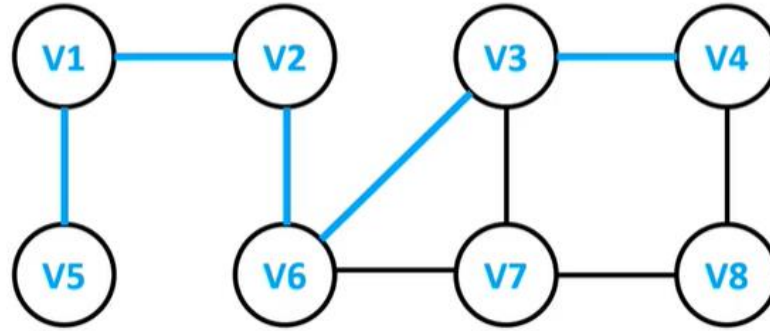
■ כמו בשאלה הקודמת, רק שבכל איטרציה נסמן או נשמור את הקודקודים שבכל רכיב



▪ קוטר של גרף

▪ קוטר בגרף הינו המרחק המקסימלי שקיים בין שני קודקודים בגרף.

▪ לדוגמא: בהינתן הגרף



▪ המרחק המקסימלי בו הוא 5

▪ אלגוריתם למציאת קוטר (1):

- $diam = 0$
- for each s in V :
 - call $BFS(G,s)$
 - $max = \text{find max value in } d$
 - if $max > diam$:
 - $diam = max$
- return $diam$

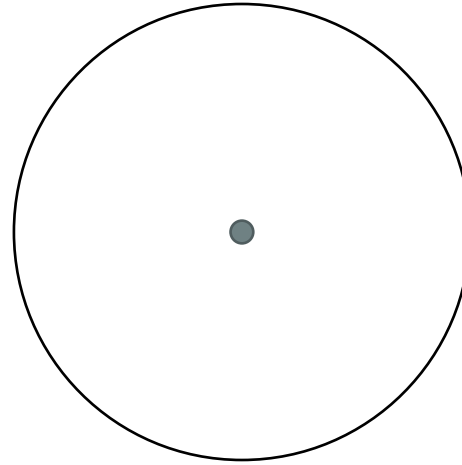
סיבוכיות:

$$O(|V| * |V+E|)$$



▪ קוטר של גרף

▪ אלגוריתם נוסף נובע מגיאומטריה במישור:



▪ אלגוריתם למציאת קוטר (2):

- select s from V
- call $\text{BFS}(G, s)$
- $u = \text{find the vertex with max value in } d$
- call $\text{BFS}(G, u)$
- return max value in d

סיבוכיות:

$O(|V+E|)$



אז מה צריך לתכנת?

■ כל מה שדיברנו עליו היום ☺

1. לממש BFS
2. האם הגרף קשיר?
3. כמה רכיבי קשירות יש?
4. מיהם הקודקודים בכל רכיב קשירות?
5. מה המרחק הקצר ביותר בין 2 קודקודים נתונים? ומה המסלול ביניהם?
6. מציאת קוטר של גרף
7. מציאת דוגמא למסלול באורך הקוטר

בהצלחה ☺

