

**Q1. (20%)**

Prove: in an undirected graph, with more than one node, there are two nodes with the same number of neighbors.

הוכיחי: בגרף לא מכוון, בעל יותר מצמת אחד, יש שני צמתים בעלי אותו מספר שכנים.

תשובה:

נניח שלכל צמת בגרף יש מספר שונה של שכנים. מכיוון שמספר השכנים האפשרי בגרף הוא בין 0 ל- $m-1$ , אז בוודאי יש בגרף צמת שאין לו שכנים (נקרא לו  $v$ ) וצמת שיש לו  $m-1$  שכנים (נקרא לו  $u$ ). כל הצמתים הם שכנים של  $u$ , ולכן  $v$  שכן של  $u$ . מצד שני ל- $v$  אין שכנים ולכן  $u$  אינו שכן של  $v$ . הגענו לסתירה, ומכאן שחייבים להיות שני צמתים בעלי אותו מספר שכנים.

**Q2. (20%)**

Describe an algorithm that finds the shortest path between two node, in an undirected connected graph with weights on the edges that are integers smaller than  $\sqrt{\log m}$  ( $m$  is the number of edges in the graph).

The running time of the algorithm is  $O(m\sqrt{\log m})$ .

תארי אלגוריתם שמוצא מסלול קצר ביותר בין שני צמתים בגרף קשיר ולא מכוון שהמשקלים של קשתותיו שלמים קטנים מ- $\sqrt{\log m}$  ( $m$  הוא מספר הקשתות בגרף).  
זמן הריצה של האלגוריתם הוא  $O(m\sqrt{\log m})$ .

תשובה:

נחלק כל קשת למספר קשתות לפי משקלה. קשת במשקל  $W$ , נחלק ל- $W$  קשתות. נריץ BFS על הגרף שנוצר.  
גודל הגרף החדש הוא  $O(m\sqrt{\log m})$  ולכן זהו גם זמן הריצה של BFS.

**Q3. (20%)**

A sink in a directed graph is a node with no outgoing edges, and  $n-1$  incoming edges ( $n$  is the number of nodes in the graph).

Write in pseudocode an algorithm that returns true if, and only if, there is a sink in a given directed graph that is represented in a matrix.

The running time of the algorithm is  $O(n)$ .

בור בגרף מכוון הוא צמת שלא יוצאות ממנו קשתות, אבל נכנסות אליו  $n-1$  קשתות ( $n$  הוא מספר הצמתים בגרף).

כתבי אלגוריתם, בפסידוקוד, אלגוריתם שמחזיר true אם ורק אם יש בור בגרף מכוון נתון. הגרף מיוצג במטריצה.

זמן הריצה של האלגוריתם הוא  $O(n)$ .

תשובה:

בסיום הלולאה הראשונה יש ב- $r$  את הצמת היחיד שיכול להיות בור.

הלולאה השנייה בודקת שהוא אכן בור.

הלולאה הראשונה רצה עמודה עמודה. אם היא מגלה *false* זה אומר שצמת  $c$  לא יכול להיות בור. אם היא מגלה *true* זה אומר שצמת  $r$  לא יכול להיות בור. לכן בסוף ריצתה יש ב- $r$  את המועמד היחיד להיות בור.

sink(G)

```
r ← 1
for c ← 2 to n do
    if G[r,c]
        then r ← c
for c ← 1 to n do
    if G[r,c] or (r ≠ c and not G[c,r])
        then return false
return true
```

**Q4. (20%)**

Prove or give a counterexample: in any undirected connected graph there is a node whose removal from the graph will leave the graph connected.

הוכיחי או תני דוגמה נגדית: בכל גרף לא מכוון וקשיר יש צמת שהוצאתו מהגרף תשאיר את הגרף קשיר.

תשובה:

נריץ BFS על הגרף ונקבל את עץ ה-BFS. בעץ זה יש מסלול מכל צמת לכל צמת בגרף (כי הגרף קשיר). אם ננתק מהעץ את אחד מעליו העץ עדיין ישאר קשיר. כלומר: אם ננתק מהגרף את הצמת הזה הגרף עדיין יהיה קשיר.

**Q5. (20%)**

Describe an algorithm that given a weighted undirected graph, finds the lightest set of edges whose removal will leave the graph connected and acyclic.

The running time of the algorithm is  $O(m \log n)$ .  $m$  is the number of edges in the graph, and  $n$  is the number of nodes in the graph.

תארי אלגוריתם שמקבל גרף לא מכוון וממושקל ומוצא את קבוצת הקשתות הקלה ביותר שיש להוציא מן הגרף כדי לקבל גרף קשיר וחסר מעגלים.  
זמן הריצה של האלגוריתם הוא  $O(m \log n)$ .  $m$  הוא מספר הקשתות בגרף, ו- $n$  הוא מספר הצמתים בגרף.

תשובה:

יש למצוא עץ פורש מקסימלי בגרף. קבוצת הקשתות שאינה בעץ היא הקבוצה המבוקשת.