נתון גרף G = (V,E) לא מכוון עם משקלים אי שליליים על הקשתות.

תארי אלגוריתם שסיבוכיותו $O(E^2\log V) = O(E(E\log V))$ המוצא מעגל קצר ביותר בגרף G.

Describe an algorithm that finds a minimum weighted cycle in a positive weighted, undirected graph G = (V,E).

Time complexity $O(E^2\log V) = O(E(E\log V))$.

שאלה 2 (25% - 10% לסעיף א'; 15% לסעיף ב')

סעיף א (10%):

הגדרה: **דרגת הכניסה** של צומת $v \in V$ בגרף מכוון G=(V,E) היא מספר הקשתות הנכנסות אל v.

בהינתן ייצוג של גרף מכוון (ולא ממושקל) על ידי מטריצת סמיכויות (adjacency matrix), מהי סיבוכיות הזמן הנדרשת לחישוב דרגת הכניסה של צומת בגרף? הסבירו.

The **indegree** of a vertex *v*, $v \in V$, in a directed graph G=(V,E), is the number of edges that "enter" vertex *v*.

Given an adjacency matrix representation of a directed (unweighted) graph, what is the time complexity required to compute the indegree of a vertex in the graph?

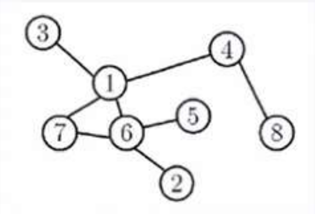
סעיף ב (15%)

זמן נותר 3:55:10

הגדרות:

צומת מפרק בגרף לא מכוון וקשיר G = (V, E) הוא צומת שכאשר מסירים אותו (ואת כל הקשתות הסמוכות לו) מהגרף, הגרף הופך ללא קשיר.

דוגמה: בגרף שלהלן כל אחד מהצמתים 1, 4 ו-6 הוא צומת מפרק בפני עצמו. אחרי שנסיר אחד מהם (ואת כל הקשתות הסמוכות לו), הגרף שיתקבל יהפוך ללא קשיר:



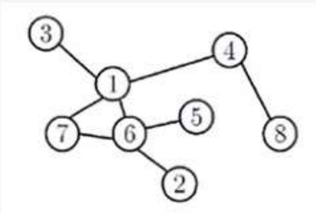
גשר בגרף לא מכוון וקשיר G = (V, E) הוא קשת שכאשר מסירים אותה מהגרף, הגרף הופך ללא קשיר.

הוכיחו או תנו דוגמה נגדית:

נתון גרף *G* קשיר ולא מכוון. אם אין ב-*G* גשרים, אזי אין ב-*G* צמתים מפרקים.

A cut vertex in an undirected connected graph G=(V,E) is a vertex that by removing it (and the edges incident to it) from graph G, disconnects the graph.

For example: in the following graph, vertices 1, 4, and 6 are each a cut vertex. Removing each one of them (and all its incident edges), will create an unconnected graph.



A bridge, in an undirected, connected graph, G=(V,E), is an edge, that when removed from the graph, it disconnects the graphs.

זמן נותר 3:53:16

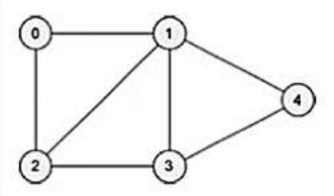
שאלה 3 (25%)

נתון גרף לא מכוון ולא ממושקל $G=(V,E)$, נתון צומת $s\in V$.

קשת (u,v) בגרף תיקרא **מיוחדת** עבור צומת s אם המרחק מ-s ל-u שווה למרחק מ-s ל-v.

תזכורת: המרחק בין x ל-y, מוגדר כאורך המסלול הקצר ביותר בין x ל-y.

למשל, עבור הגרף להלן והצומת 0, הקשת (3,4) היא מיוחדת, שכן המרחק הקצר ביותר מצומת 0 לצומת 3 שווה למרחק הקצר ביותר מצומת 0 לצומת 4 (ושווה ל- 2).



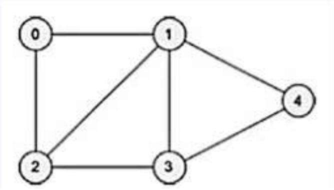
נתון גרף לא מכוון $G=(V,E)$ ונתון צומת s, $(s\in V)$. תארי אלגוריתם, יעיל ככל הניתן, המוצא את כל הקשתות המיוחדות ביחס ל-s בגרף G. הניחי שהגרף מיוצג על ידי רשימת סמיכויות (adjacency list) ונתחי את זמן הריצה של האלגוריתם שתיארת.

Given an undirected, unweighted graph G = (V,E), and a vertex s, $s\in V$.

Edge (u, v) in the graph, will be called **special** in relation to vertex s, if the distance from s to u is equal to the distance from s to v.

Reminder: the distance between x to y, is defined as the length of the shortest path between x and y.

For example: for the following graph, and vertex 0, edge (3,4) is special, since the distance between vertex 0 and vertex 3 is equal to the distance between vertex 0 and vertex 4 (and is equal to 2).



שאלה 4 (25%)

נתון גרף קשיר ממושקל ולא מכוון $G=(V,E)$ וקבוצת קשתות K $(K\subseteq E)$ בגרף, הצבועות בצבע ירוק.

תארי אלגוריתם המוצא בזמן **O(E lg$V$)** עפ"מ (MST) ירוק ביותר של G , כלומר עץ פורש מינימלי (Minimum Spanning Tree) של G המכיל מספר רב ככל הניתן של קשתות ירוקות.

Given a connected, weighted, undirected graph $G=(V,E)$ and a group of edges K, $(K\subseteq E)$, which are coloured as green.

Describe and algorithm, that finds the greenest minimum spanning tree of graph G. In other words, find the minimum spanning tree of graph G, that includes the largest number of green vertices.

The algorithms time complexity should be: **O(E lg$V$).**

שאלה 5 (25%: 10% לסעיף א׳; 15% לסעיף ב׳)

סעיף א׳ (10%)

האם הטענה הבאה נכונה?

אם כן - נמקי. אחרת, תני דוגמה והסבירי במפורט מדוע הטענה אינה נכונה:

הטענה:

בגרף קשיר, לא מכוון שיש בו קשתות שליליות, האלגוריתם של דייקסטרה נכשל **תמיד** במציאת מסלול קצר ביותר בין שני צמתים בגרף.

Is the following statement true or false?

If true, explain why. Otherwise, give an example and explain (in words) why the statement is incorrect:

In a connected, undirected graph that has negative arcs, Dijkstra's algorithm will **always** fail to find a shortest path between two nodes in the graph.

סעיף ב׳ (15%)

יהי G = (V,E) גרף קשיר ממושקל ולא מכוון כך שאין ב- *G* מעגלים שליליים. נתון כי s צומת ב-*G*, $(s \in V)$.

הגדרה:

גרף מסלולים זולים ביותר מ-*s* של *G*, הוא עץ לא מכוון, ***PG***, אשר בו לכל צומת $v \in V$ בגרף, המסלול היחיד מ-*s* ל-*v* ב-*PG* הינו מסלול קל ביותר מ-*s* ל-*v* ב-*G*.

הוכיחי או תני דוגמה נגדית:

אם *T* הוא גרף מסלולים זולים ביותר מ- s של *G* אזי *T* הוא עפ"מ (*MST*) של *G*.

תזכורת: עפ"מ = עץ פורש מינימלי.

Graph G=(V, E) is a connected, weighted, undirected graph, which does not have negative circular paths. *s* is a vertex in graph G, $(s \in V)$..

Definition:

A ***cheapest path graph* from s in G**, is an undirected tree, PG, in which, for each vertex *v*, $v \in V$, the only path from s to v in PG, is a minimum weight path from s to v in G.

Prove or give a counter example:

If T is a cheapest path graph from s in graph G, then T is a minimum spanning tree (MST) of G.