

שם: דניאל בספלוב

תעודת זהות: 213446479

Q1- קובץ קוד של הקוד שניסיתי לפתור את התרגיל באמצעות מערך אך לא נבחר בסוף.

Q2- פתרון אותו תרגיל כמו בשאלה 1 אך עם רשימה מקושרת. הסבר למה נבחר דווקא פתרון זה בהמשך הקובץ.

כל השאר זה לפי השלבים במטלה. הקליינט שאיתו מתקשרים אותו אותו קליינט אך רק הסרבר משתנה. למשל השרת בשלב 4 נקרא q4server. כל הקבצים מתקמפלים במייק

שלב 2:

תחילה אתחיל כי קודם קבעתי באיזה שיטה אשתמש (סעיף 2.2) ואז באיזה שיטה להשתמש ברשימה מקושרת (2.1) בחרתי לממש באמצעות שתי דרכים.

הדרך הראשונה היא מערך דו מימדי;

המעריך בגודל כמות הקשתות וכל מערך "פנימי" מכיל שתי מקומות שמייצגים מאיזה מספר לאיזה מספר הקשת.

אני מריץ את הקוד עם אלף קודקודים ואלף קשתות

```
danielbes@Ubuntu:~/Desktop/OS_EX3$ ./Q1
Write please num of verts and after num of arcs:
10000 10000
```

לאחר מכן מעביר את הנתונים של הפרופילנג לקובץ טקסט (analysisist)

```
gprof Q1 gmon.out > analysis.txt
```

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self us/call	total us/call	name
53.85	0.21	0.21	10000	21.00	21.00	printSCCs
46.15	0.39	0.18	10000	18.00	18.00	DFS
0.00	0.39	0.00	1	0.00	0.00	reverseGraph

עכשיו אשתמש בדרך השנייה, רשימה מקושרת.

אני מריץ שוב עם קלט גדול, 10000 קודקודים ו10000 קשתות

```
danielbes@Ubuntu:~/Desktop/OS_EX3$ ./q2
Enter the number of vertices and edges: 10000 10000
```

לאחר מכן מעביר את הנתונים של הפרופילינג לקובץ טקסט (analysisist2)

```
danielbes@Ubuntu:~/Desktop/OS_EX3$ gprof q2 gmon.out > analysis2.txt
```

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
100.00	0.01	0.01				main
0.00	0.01	0.00	20000	0.00	0.00	addEdge
0.00	0.01	0.00	20000	0.00	0.00	newAdjListNode
0.00	0.01	0.00	9935	0.00	0.00	printSCCUtil
0.00	0.01	0.00	5076	0.00	0.00	DFSutil
0.00	0.01	0.00	1	0.00	0.00	getTranspose

קל לראות, שבאמצעות רשימה מקושרת הפונקציות פועלות יותר מהר לכן ממשיך עם רשימה מקושרת.