
תכנות מערכות 1

סמסטר א תשפד

מטלה 2 - לולאות ומערכים

- המטלה היא בזוגות או ביחידים – לא שלשות.
- יש לקמפל עם הדגלי אזהרות שנלמדו בתרגולים, על מנת לוודא שהתוכנית מתקמפלת ללא אזהרות. מצב שתוכנית מתקמפלת עם אזהרות יגרור הורדת נקודות!
- יש לוודא שהתוכנית מתקמפלת על מערכת ההפעלה של לינוקס UBUNTU עם קומפיילר שהתקנו בתרגול הראשון.
- יש להגיש את המטלה ב git , ל MOODLE יש להגיש קובץ txt :
 1. שורה ראשונה תכלול את כתובת ה git (לא URL!)
 2. שורה שנייה תכלול את מזהה ה commit הרלוונטי
 3. שורה שלישית תכיל את תעודת הזהות של הסטודנטים המגשים מופרדים ברווח
- הינכם נדרשים לקוד קריא ונקי.
- בכל מקום בו יש צורך בשימוש בקבועים בעלי משמעות יש להגדיר אותם באמצעות **define**

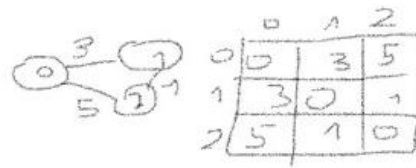
שאלה 1

- **מטריצת שכנויות:** היא שיטה לייצוג גרף מכוון / לא מכוון בעל N צמתים (קודקודים)
 - בעזרת מטריצה ריבועית, $N \times N$ שערכי תאיה הם 0 או מספר חיובי. תא (i, j) בגרף מתאר את משקלה של הקשת מקודקוד i לקודקוד j בגרף.
 - אם אין קשת כזו, הערך בתא במטריצה יהיה 0.
 - אם יש קשת כזו, הערך יהיה מספר חיובי והוא ייצג את משקל המעבר מקודקוד i לקודקוד j.
- עליכם לבנות תוכנית המייצרת בעזרת מטריצה גרף לא מכוון ומבצעת את הפעולות הבאות:
 - א. התוכנית קולטת מהמשתמש את ערכי המטריצה (לא דווקא בתחילת התוכנית. יכול להיות מצב בו אתם מקבלים מטריצה מבצעים בדיקות ואז מבלי לצאת מהתוכנית אתם נדרשים לקבל עוד מטריצה)
 - ב. התוכנית קולטת מהמשתמש את ערכי i ו j ומדפיסה True אם קיים מסלול מ i ל j . אחרת התוכנית תדפיס False .
 - ג. התוכנית קולטת מהמשתמש את ערכי i ו j ומדפיסה את המסלול הקצר ביותר מ i ל j . אם לא קיים מסלול התוכנית תדפיס מינוס 1.

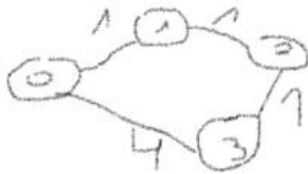
יש להניח כי הקלט תקין.
גודל המטריצה הוא 10 על 10

שימו לב- לשאלה זו סופקו דוגמאות קלט-פלט חובה על הפלט שלכם להיות זהה לפלט שסופק.

• דוגמא:



מסלול הקצר ביותר מקודקוד 0 לקודקוד 2 הוא: 4



	0	1	2	3
0	0	1	0	4
1	1	0	1	0
2	0	1	0	1
3	4	0	1	0

מסלול הקצר ביותר מקודקוד 0 לקודקוד 3 הוא: 3

אלגוריתמים העוזרים לכם לפתור בעיה זו:

Floyd-Warshall algorithm (עדיף)

אלגוריתם פלוייד-וורשאל הוא אלגוריתם במדעי המחשב המשמש למציאת המסלולים הקצרים ביותר בין כל שני צמתים, בגרף ממושקל ומכוון/ לא מכוון

האלגוריתם מתבסס על האבחנה הבאה: אם ממספרים את קבוצת הצמתים של הגרף כך שהיא מסומנת $\{1, 2, \dots, n\}$ ומסמנים בתור k את משקל המסלול הקצר ביותר מהצומת i אל הצומת j שעובר בצמתי ביניים השייכים אך ורק לקבוצה $\{1, 2, \dots, k\}$ אז מתקיים:

$$(d(i,j))^k = \min\{(d(i,j))^{(k-1)}, (d(i,k))^k + (d(k,j))^k\}$$

פעולת האלגוריתם היא כך: הוא מבצע $|V|$ איטרציות כאשר באיטרציה k הוא מחשב את $(d(i,j))^k$ לכל זוג צמתים i, j השייכים ל V - ושומר את המידע בטבלה. מידע נוסף על האלגוריתם ניתן למצוא בקישורים הבאים:

https://he.wikipedia.org/wiki/%D7%90%D7%9C%D7%92%D7%95%D7%A8%D7%99%D7%AA%D7%9D_%D7%A4%D7%9C%D7%95%D7%99%D7%93-%D7%95%D7%95%D7%A8%D7%A9%D7%90%D7%9C

<https://www.youtube.com/watch?v=ONi0Rf2P9GE>

אופציה נוספת - dykstra algorithm:

www.wikipedia.org ויקיפדיה - אלגוריתם דייקסטרה

עליכם לכתוב ספריה בשם **my_mat** אשר תכיל פונקציה עבור כל פעולה. התוכנית הראשית תקרא מהשתמש את הפעולות אחת אחת ותבצע כל פעולה על ידי קריאה לפונקציה המתאימה. ביצוע פעולות מסוימות ידרוש קליטת נתונים נוספים והעברתם לפונקציה המבצעת את הפעולה. על התוכנית הראשית לקרוא פעולות עד שהשתמש יבחר להפסיק פעולה.

שם קובץ ההרצה של התוכנית הראשית my_graph
המשתמש יקליד כל פעם אחת מאותיות אלו כדי לבצע את הפונקציות הבאות:
A – ביצוע פונקציה 1
B – ביצוע פונקציה 2
C – ביצוע פונקציה 3
EOF או D – יציאה מהתוכנית

שאלה 2

- א. גשו לדף של בעיית תרמיל הגב בשלמים בוויקיפדיה וקראו על הבעיה [תרמיל הגב בוויקיפדיה](#)
- ב. כתבו תוכנית שיוצרת רשימה של 5 פריטים, נותנת משקל וערך לכל פריט (קלט מהמשתמש).
- ג. כתבו פונקציה שקובעת אילו פריטים לכלול בתיק כך שהמשקל הכולל קטן או שווה ל-20 קילוגרם והערך הכולל יהיה מקסימלי **ותחזיר את הערך המקסימלי של פריטים שנכנסו לתרמיל.**
- ד. **התוכנית הראשית בשם my_Knapsack** תדפיס את רשימת הפריטים שהתיק יכול שיענו על הדרישות ואת הערך הכולל המקסימלי.
- ה. **משקל התיק וגודל המערך צריכים להיות מוגדרים באמצעות define.**
- ו. **מערך result יוגדר להיות בגודל מקסימלי בפונקציית main.**
- ז. **הכנסת המוצרים למערך result יהיה לפי סדר הופעתם במערך**

חתימת הפונקציה :

```
int knapSack (int weights[], int values[] , int selected_bool[])
```

דוגמאות לקלט ופלט:

Example 1 Input:

```
items = [A,B,C,D,E]
```

```
values = [35, 60, 100, 120, 130]
```

```
weights = [2, 5, 6, 9, 10]
```

Example 1 Output:

Maximum profit: 280

Items that give the maximum profit: [D, C, B]

Example 2 Input:

items = [A,B,C,D,E]

values = [35, 60, 100, 120, 130]

weights = [2, 5, 6, 9, 9]

Example 2 Output:

Maximum profit: 290

Items that give the maximum profit: [E, C, B]

Example 3 Input:

items = [A,B,C,D,E]

values = [35, 60, 100, 120, 130]

weights = [2, 5, 6, 9, 7]

Example 3 Output:

Maximum profit: 325

Items that give the maximum profit: [E, C, B, A]

שימו לב – לשאלה זו ניתנו דוגמאות בלבד לקלט פלט תקינים, אין לשאלה זו קבצי פלט קלט לבדיקה.

תוצרים:

עליכם לכלול את הקבצים הבאים, כולם באותה תיקייה ראשית ללא תתי-תיקיות:

- Makefile
- my_mat.h
- my_mat.c
- **my_graph.c**
- my_Knapsack.c

שמות קבצי ההרצה שאמורים להיווצר הם:

my_graph - התוכנית לשאלה 1

my_Knapsack - התוכנית שאלה 2

בהצלחה!