

## תרגיל 4:

1. יצרנו קובץ cpp בשם `dijkstra.cpp` הקוד ברובו מבוסס על הקוד מהאתר אותו התבקשנו לעדכן. הוספנו מספר בדיקות לשגיאות הסותרות את היגיון האלגוריתם, וכן בדיקות לתקינות קלט לפי הפורמט להכנסת גרף אותו הצגנו בקוד.
2. יצרנו קובץ `Makefile`, אשר יוצר את קובץ ההרצה ומריץ את התוכנית בפעולה אחת (`make run`). בעת ההרצה ישנם דגלי קומפילציה אשר דואגים לכך שהקוד "ימופה" בכדי שנוכל להשתמש ביכולות `gcov` לדעת את כיסוי הקוד.  
הוראות קימפול:

```
nitzan@nitzan-VirtualBox: ~/Desktop/OS_EX1/q4
nitza@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$ make run
g++ -Wall -Wextra -g -fprofile-arcs -ftest-coverage -o dijkstra dijkstra.cpp
./dijkstra

Welcome to the Dijkstra program.
Input example:
  First line: number of vertices V and number of edges E
  Next E lines: each line contains edge in the format: u v w
  Last line: source vertex
Example input:
5 6
0 1 2
1 2 3
0 2 6
1 3 1
3 4 5
2 4 2
0

Enter number of vertices and edges (or Ctrl+D to exit):
```

בעת הקומפילציה, ע"י `make run`, הקובץ מתחיל לרוץ. מוצג הפורמט כיצד להכניס קלט של גרף.

## דוגמא לגרף תקין:

```
Enter number of vertices and edges (or Ctrl+D to exit):
5 6
Enter edges (u v w) each in separate line:
0 1 2
0 2 6
1 3 1
3 4 5
2 4 2
1 2 3
Enter source vertex:
0
Shortest distances from vertex 0:
0 2 5 3 7
Enter number of vertices and edges (or Ctrl+D to exit):
```

## גריץ gcov:

```
nitzan@nitzan-VirtualBox:~/Desktop/05_EX1/q4$ gcov dijkstra.cpp
File 'dijkstra.cpp'
Lines executed:82.09% of 67
Creating 'dijkstra.cpp.gcov'

File '/usr/include/c++/11/iostream'
No executable lines
Removing 'iostream.gcov'

File '/usr/include/c++/11/bits/stl_algobase.h'
Lines executed:94.29% of 70
Creating 'stl_algobase.h.gcov'

File '/usr/include/c++/11/bits/predefined_ops.h'
Lines executed:85.71% of 14
Creating 'predefined_ops.h.gcov'

File '/usr/include/c++/11/bits/stl_iterator.h'
Lines executed:100.00% of 24
Creating 'stl_iterator.h.gcov'

File '/usr/include/c++/11/bits/stl_uninitialized.h'
Lines executed:100.00% of 36
Creating 'stl_uninitialized.h.gcov'

File '/usr/include/c++/11/ext/new_allocator.h'
Lines executed:84.21% of 19
Creating 'new_allocator.h.gcov'

File '/usr/include/c++/11/bits/move.h'
Lines executed:100.00% of 6
Creating 'move.h.gcov'

File '/usr/include/c++/11/bits/stl_vector.h'
Lines executed:98.73% of 158
Creating 'stl_vector.h.gcov'

File '/usr/include/c++/11/bits/stl_iterator_base_types.h'
Lines executed:100.00% of 2
Creating 'stl_iterator_base_types.h.gcov'

File '/usr/include/c++/11/bits/alloc_traits.h'
Lines executed:100.00% of 20
Creating 'alloc_traits.h.gcov'

File '/usr/include/c++/11/bits/cpp_type_traits.h'
Lines executed:100.00% of 2
Creating 'cpp_type_traits.h.gcov'

File '/usr/include/c++/11/bits/stl_construct.h'
Lines executed:100.00% of 14
Creating 'stl_construct.h.gcov'

File '/usr/include/c++/11/bits/stl_function.h'
Lines executed:100.00% of 2
Creating 'stl_function.h.gcov'

Lines executed:94.76% of 534
```

ניתן לראות כיסוי של 94.76% כלומר יש מספר שורות שלא כוסו. נבדוק אילו שורות לא כוסו ע"י בדיקת הקובץ שנוצר בשם `dijkstra.cpp.gcov`. הרצנו ע"י `grep "####"` + שם הקובץ, בדיקה על הקובץ אשר מציגה לנו את כל השורות שלא כוסו. ניתן לראות שמכיוון שקלט הגרף היה תקין, לא נכנסנו לשגיאות בקוד. ולכן חלקים אלו לא כוסו.

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$ grep "####" dijkstra.cpp.gcov
#####: 19:      cerr << "Error: Dijkstra does not support negative weights.\n";
#####: 20:      exit(1);
#####: 25:      cerr << "Error: Invalid vertex index in edge.\n";
#####: 26:      exit(1);
#####: 89:      cerr << "Error: Invalid number of vertices or edges.\n";
#####: 90:      continue;
#####: 98:          cerr << "Error: Invalid edge input.\n";
#####: 99:          return 1;
#####: 108:      cerr << "Error: Invalid source vertex.\n";
#####: 109:      return 1;
#####: 117:      cout << "INF ";
```

כעת נציג מקרים בהם הכנסנו קלט גרף שגוי במכוון, בכדי לבדוק שהקוד מכוסה במקרים אלו:

### בדיקת משקל שלילי לקשת:

הכנסו קלט גרף לא תקין (משקל 2- לקשת):

```
Enter number of vertices and edges (or Ctrl+D to exit):
3 1
Enter edges (u v w) each in separate line:
0 1 -2
Enter source vertex:
0
Error: Dijkstra does not support negative weights.
make: *** [Makefile:12: run] Error 1
```

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$ grep "####" dijkstra.cpp.gcov
#####: 25:      cerr << "Error: Invalid vertex index in edge.\n";
#####: 26:      exit(1);
#####: 89:      cerr << "Error: Invalid number of vertices or edges.\n";
#####: 90:      continue;
#####: 98:          cerr << "Error: Invalid edge input.\n";
#####: 99:          return 1;
#####: 108:      cerr << "Error: Invalid source vertex.\n";
#####: 109:      return 1;
#####: 117:      cout << "INF ";
#####: 124:      return 0;
```

ניתן לראות שכיסוינו הפעם את שורות 19-20 שעסקו בחריגה זו.

### בדיקת קשת מחוץ לטווח:

הגדרנו שיש 3 ק', והכנסנו קשת מ-9 ל-0. חריגת טווח.

```
Enter number of vertices and edges (or Ctrl+D to exit):
3 1
Enter edges (u v w) each in separate line:
0 9 3
Enter source vertex:
0
Error: Invalid vertex index in edge.
make: *** [Makefile:12: run] Error 1
```

נבדוק בgcov:

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$ grep "#####" dijkstra.cpp.gcov
#####: 89:      cerr << "Error: Invalid number of vertices or edges.\n";
#####: 90:      continue;
#####: 98:      cerr << "Error: Invalid edge input.\n";
#####: 99:      return 1;
#####: 108:     cerr << "Error: Invalid source vertex.\n";
#####: 109:     return 1;
#####: 117:     cout << "INF ";
#####: 124:     return 0;
```

אכן שורות 25-26 שעסקו בחריגה זו, כוסו הפעם.  
**בדיקת גרף לא חוקי (0 קודקודים, 2 קשתות):**

```
Enter number of vertices and edges (or Ctrl+D to exit):
0 2
Error: Invalid number of vertices or edges.
```

נלחץ ctrl-d כדי לצאת מהלולאה לאחר השגיאה.

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$ grep "#####" dijkstra.cpp.gcov
#####: 98:      cerr << "Error: Invalid edge input.\n";
#####: 99:      return 1;
#####: 108:     cerr << "Error: Invalid source vertex.\n";
#####: 109:     return 1;
#####: 117:     cout << "INF ";
```

ניתן לראות שגם שורות 89-90 כוסו!

**בדיקת קלט שגוי לקשת (שני ערכים במקום 3):**

```
Enter edges (u v w) each in separate line:
0 1 4
1 2 x
Error: Invalid edge input.
```

הכנסנו ערך x המדמה הכנסת 2 ערכים / קלט לא חוקי של פחות מ3 משתנים.

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$ grep "#####" dijkstra.cpp.gcov
#####: 108:     cerr << "Error: Invalid source vertex.\n";
#####: 109:     return 1;
#####: 117:     cout << "INF ";
```

ניתן לראות ששורות 98-99 כוסו בהצלחה.

**בדיקת קודקוד מקור לא תקין:**

```
Enter number of vertices and edges (or Ctrl+D to exit):
3 1
Enter edges (u v w) each in separate line:
0 1 4
Enter source vertex:
5
Error: Invalid source vertex.
```

הגדרנו גרף עם 3 ק', ובחרנו את ק' 5 בתור מקור. לא חוקי. נבדוק gcov כעת:

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$ grep "#####" dijkstra.cpp.gcov
#####: 117:     cout << "INF ";
```

גם שורות 108-109 מכוסות כעת. נותר לנו לבדוק את שורה 117.



### בדיקת צומת מנותק (INF):

```
Enter number of vertices and edges (or Ctrl+D to exit):
3 1
Enter edges (u v w) each in separate line:
0 1 3
Enter source vertex:
0
Shortest distances from vertex 0:
0 3 INF
```

הכנסנו 3 ק', חיברנו בין 0-1, והרצנו דייקסטרה מק' 0.  
ק' 2 מנותק מק' 0. כלומר האלגוריתם מציג את המרחק ביניהם כ-INF.

נריץ gcov:

```
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$ grep "#####" dijkstra.cpp.gcov
nitzan@nitzan-VirtualBox:~/Desktop/OS_EX1/q4$
```

ניתן לראות שכעת בעת חיפוש "#####" בקוד, לא נותרו אזורי קוד לא  
מסומנים. כלומר נכנסנו לכל האזורים בקוד.