

CPSC 319 – Summer 2020 Assignment 3 <b>Graphs</b> (10% of total final grade)	Released:	August 4 (Tuesday) @ 10:00 AM
	Due:	August 21 (Friday) @ 10:00 AM
	# days to complete:	17 days (no extensions)

## GOAL

Graphs are invaluable data structures. You will be given a Java source code that implements Dijkstra's Shortest Path algorithm and prints the resulting paths. In this assignment, you will write a parser to read a text file to construct an adjacency matrix graph used in the given Java code.

## INSTRUCTIONS

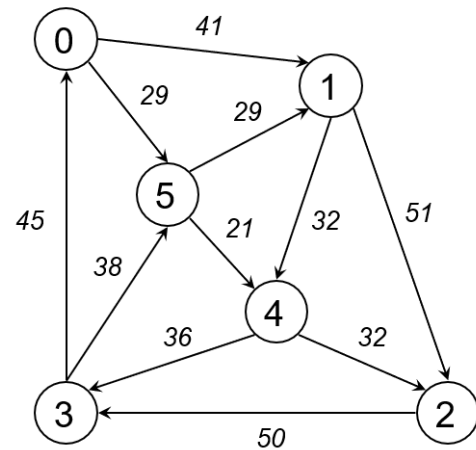
Extend the given Java source code "DijkstrasAlgorithm" to perform the following tasks:

### 1. Request the user for the name of a text file.

Example input text file "ex6"

```
# ex6.txt
#
0 41 0 0 0 29
0 0 51 0 32 0
0 0 0 50 0 0
45 0 0 0 0 38
0 0 32 36 0 0
0 29 0 0 21 0
```

This ex6.txt file represents an adjacency matrix structuring the graph on the right discussed in class:



### 2. Write a parser to read the input file and adapt it to the given Java code. Your parser will initialize the adjacency matrix after the input text file given by the user (i.e., yellow highlighted text below)

```
class Asgmt3 {
    public static void main(String[] args) {
        int[][] matrix = AdjacencyMatrixParser.parseFile(inputFileName);

        DijkstrasAlgorithm.dijkstra(matrix, 0);
    }
}
```

**Example** of the resulting 6x6 matrix after parsing the input text file `inputFileName` = "ex6 "

```
int[][] matrix =
```

	0	1	2	3	4	5
0	0	41	0	0	0	29
1	0	0	51	0	32	0
2	0	0	0	50	0	0
3	45	0	0	0	0	38
4	0	0	32	36	0	0
5	0	29	0	0	21	0

3. Write the output of the given Java code to a text file. Note the given Java code displays the output on the screen. Your program should explicitly write the output to the text file with name "`inputFileName_dijkstra_output`"

**Example** of content of output file name "ex6\_dijkstra\_output"

Vertex	Distance	Path
0 -> 1	41	0 1
0 -> 2	82	0 5 4 2
0 -> 3	86	0 5 4 3
0 -> 4	50	0 5 4
0 -> 5	29	0 5

### GIVEN CODES & TEXT FILES

1. **Asgmt3.java** – You should adapt it to accept the input file name (Step 2)
2. **DijkstrasAlgorithm.java** – Source code you should modify for the assignment
3. **FileIO.java** – You should adapt it in your program
4. **Four input files** representing cities and distances between them (you should process) and corresponding output (for your reference)

Input Text Files	Reference Output Text Files (i.e., your program must match them)
sp11	sp11_dijkstra_output
ha30	ha30_dijkstra_output
wg59	wg59_dijkstra_output
sgb128	sgb128_dijkstra_output

**HAND-IN**

1. Your code for **Asgmt3.java**, **AdjacencyMatrixParser.java**, **FileIO.java**
2. README (w/ instructions on how to run and use/interact with your program)
3. Please include all of the above items in a zipped folder titled (asgmt3-your LAST NAME-ID).zip and submit it to D2L dropbox.

**LATE ASSIGNMENTS WILL NOT BE ACCEPTED – START WORKING IN YOUR ASSIGNMENT EARLY**

**MARKING**

**Source code that does not compile or produces run-time errors will receive a grade of 0%.**

Item		Points
1	<b>Request the user for the name of a text file</b>	<b>1 point</b>
2	<b>Correct output format and content for the 4 given input text files.</b> The corresponding output files are given for your reference.	<b>4 points</b>
<i>Total</i>		<b>5 points</b>

**INDIVIDUAL WORK**

- The assignment must be done **individually**, so you must write up the solutions *on your own* in *your own words*.
- Everything that you hand in must be your original work, except for the code copied from the textbook, lecture material (i.e., slides, notes), web, or that supplied by your TA. When someone else's code is used like this, you **must** acknowledge the source explicitly, citing the sources in a scientific way (i.e., including author(s), title, page numbers, URLs, etc.).
- Copying another student's work constitutes academic misconduct, a severe offence that will be dealt with rigorously in all cases. Please read the sections of the University Calendar under the heading "Student Misconduct."
- If you are in doubt whether a specific form of aid is allowed, ask your instructor!
- Contact your TA if you have problems getting your code to work.
- Note that your code may be checked thoroughly for plagiarism by computer.

**END OF THE ASSIGNMENT**