Assignment 2 Specs

Thursday, November 4, 2021 4:01 PM

[Disclaimer]: This document is made out of goodwill, and while the author continues to try maintaining the most up-to-date information regarding ECE650 A2, **the author however is not responsible for any inaccurate information**. Please use this reference at your own risk, **You are strongly advised to refer to the actual posts and those posts answered by TAs**.

Programming Language: C++

Input:

- Your program should take input from standard input.
- There are TWO kinds of input:
  1. Specification of an undirected graph (Vertices V or Edges E): the specification of a set of vertices V, and set of edges E of the undirected graph (as the example below in "Testing Your Program" Indicates)
     i. The specification of a set of vertices **starts with 'V'**, **followed by a space**, **followed by an integer greater than one**, **all in one single line.** If the integer that follows the V is i, then **we assume that the vertices are identified by 1, . . . , i**.
     ii. The specification for a set of edges **starts with 'E'**. It **then has a space**, **followed by the set of edges in a single line delimited by '{' and '}'**. The **two vertices of an edge are delimited by '<' and '>' and separated by a comma**. The **edges in the set are also separated by a comma**. There are **no whitespace characters within the { }**.
  2. Commands that asks to print out a shortest-path from one vertex to another in the current graph
     - The only other kind of input **starts with an 's'**. It asks for a shortest path from the first vertex to the second that is specified after the s. **The s is followed by a space**, **a vertex ID**, **another space**, and **a second vertex ID**.
- **When a new V specification starts, all previous information (previous graphs) can be forgotten**. (There is no relationship between subsequent graph specifications.)
- **After the specification of the graph there can be zero or more shortest path**

- After the specification of the graph there can be zero or more shortest-path queries s.

Output:

- Your program should output to standard output
- Errors can also be output to standard output, but should always start with "Error:" followed by a brief description.
- Your program should NOT generate any extraneous output; for example, do NOT print out prompt strings such as "please enter input" and things like that.
- The lines 2-8-10 and 5-2-3-1 below in "Testing Your Program" are outputs of the s commands that immediately precede them. The output comprises vertex IDs separated by -, with no whitespace within.
- Our input will be perfectly formed (No format errors).
- For each s query, only one shortest path should be output; multiple shortest paths might exist and an arbitrary choice can be made.
- The Kinds of Error:
  - (When query for shortest path) If a path does not exist between the vertices, you should output an error.
    - when the vertices exist, but a path does not exist between them.
  - (When query for shortest path) we may ask for a shortest path to a vertex that does not exist.
    - source and destination must both exist. If not, raise an error
  - (When specifying edges) If we have V 5, we may try to specify an edge <2,10>, for a vertex 10 that does not exist.

Termination:

- The program takes input till it sees an EOF.
- Your program should terminate gracefully (and quietly) once it sees EOF.

Testing Your Program:

- Ensure that it compiles with the C++compiler on eceubuntu, and runs on eceubuntu
- Sample Run:

  Assume that your executable is called ece650-a2. In the following, "$" is the command-prompt.

  $ ./ece650-a2

  V 15 //input: to indicate that there are 15 vertices $v_1 \ldots v_i$.

  E {<2,6>,<2,8>,<2,5>,<6,5>,<5,8>,<6,10>,<10,8>} //input: specifying edge set

s 2 10 //input: to find the shortest path from v_2 to v_{10}
2-8-10. //output from the above command: shortest path from v_2 to v_{10}
V5
E {<1,3>,<3,2>,<3,4>,<4,5>,<5,2>}
s51
5-2-3-1

Additional Resource:

- A2: A guide for those who may be struggling it (credit to Al-Baraa El-Hag): https://piazza.com/class/ktachtjrz9z2c6?cid=484
- The book, "Introduction to Algorithms", by Cormen, Leiserson, Rivest & Stein, may be useful to you. The 2nd ed. is available electronically at the library via lib.uwaterloo.ca. Section 22.1 discusses how a graph may be represented. For determining a shortest-path, you can use a simple algorithm, such as Breadth-First Search (22.2), or Bellman-Ford (24.1).

Marking

Your output has to perfectly match what is expected. You should also follow the submission instructions carefully. It discusses how to name your files, how to submit, etc. The reason is that our marking is automated.

• Does not compile/make/crashes: automatic 0
• Your program runs, awaits input and does not crash on input: + 20 • Passes Test Case 1: + 20
• Passes Test Case 2: + 20
• Passes Test Case 3: + 15
• Correctly detects errors: + 20
• Programming style: + 5 (https://piazza.com/class/ktachtjrz9z2c6?cid=365) specifically, use c++ 11 standard

CMake

As discussed below under "Submission Instructions", **you should create a CMakeLists.txt file to build your project**. We will build your project using the following sequence:
cd a2 && mkdir build && cd build && cmake ../ && make

at the top level of your repository. If your code is not compiled from scratch (i.e., from the C++ sources), you get an automatic 0.

Submission Instructions

You should **place all your files at the a2 directory in your GitLab repository**. The directory should contain:

- **All your C++ source-code files; you may use further sub-directories if you wish.**
- **A CMakeLists.txt**, that builds a final executable named ece650-a2.
- **A file user.yml that includes your name, WatIAM, and student number.**

See README.md for any additional information.
The submitted files should be in a2 directory in the master branch of your repository.

FAQ:

- How to parse inputs (edges)?
  - Regex (https://piazza.com/class/ktachtjrz9z2c6?cid=352)
  - Scan using a while loop (https://piazza.com/class/ktachtjrz9z2c6?cid=321)

- Is the input case sensitive (for V, E and s)?
  - Yes, your program should only expect three letters: **(uppercase) V, E, and (lowercase) s**

- What if we input E{<1,1>} (self-loop) https://piazza.com/class/ktachtjrz9z2c6?cid=324
  - Such input will not be considered as we assume the inputs are perfect
  - We also assume "s 1 1" to be invalid

- Will there be negative sign involved? (negative vertices/edges)?
  - No, negative signs are not valid and will not be considered during testing. (https://piazza.com/class/ktachtjrz9z2c6?cid=453)

- Do we throw an error for wrong sequence of input (i.e. Specifying edges before vertex)?
  - No https://piazza.com/class/ktachtjrz9z2c6?cid=324

- Will there be multiple edge inputs? (i.e. Specifying V and Specifying E and then Specifying E again immediately)?
  - No https://piazza.com/class/ktachtjrz9z2c6?cid=324

- What if there are repeated edges in the edge set E? (i.e. E {<**1,2**>,<3,4>,<3,2>,<**2,1**>})
    - This won't be tested https://piazza.com/class/ktachtjrz9z2c6?cid=324

- How do we test our program?
    - The test.cpp is just a minimal example showing how one would go about testing a C++ program. **You would need to first come up with some example input and outputs, and then run your code on the same input, and use test.cpp to compare the outputs to see if they match (the CHECK() part).**
    - https://piazza.com/class/ktachtjrz9z2c6?cid=353

- What if clang-tidy reports warnings/errors to a line that is crucial to my program?
    - You are not supposed to remove lines of code that causes warning/error, you are supposed to figure out what is wrong with them and fix them :)
    - https://piazza.com/class/ktachtjrz9z2c6?cid=388

- What libraries are allowed for use?
    - Any libraries that is on eceubuntu
    - https://piazza.com/class/ktachtjrz9z2c6?cid=447

- If the edge set is invalid and we print an error message and then what?
    - We should reject the whole wrongly formed graph (V&E altogether). There is no point in further processing as we know the input will not work.
    - We will reject the whole graph (previously specified V and E) if something (E) is invalid and wait for a whole new set of V and E.
    - https://piazza.com/class/ktachtjrz9z2c6?cid=448

- Will we be provided with pre-test in a2?
    - Yes
    - https://piazza.com/class/ktachtjrz9z2c6?cid=449
    - https://piazza.com/class/ktachtjrz9z2c6?cid=510

- Indexing of vertices:
    - As per specification of A2: "If the integer that follows the V is i, then we assume that the vertices are identified by 1, ..., i. (https://piazza.com/class/ktachtjrz9z2c6?cid=460)

- Is V 1 an input that will be considered? If so is this a valid input? In other words, do you test if we have considered V 1 to be valid or invalid?
    - The number of vertices are always greater than 1 (https://piazza.com/class/ktachtjrz9z2c6?cid=507)
    - https://piazza.com/class/ktachtjrz9z2c6?cid=473

- How to compile A2 skeleton code?
    - https://piazza.com/class/ktachtjrz9z2c6?cid=478

- During testing, will the inputs be given in the correct order (V, then E, then s)?
    - Yes, you can assume that (https://piazza.com/class/ktachtjrz9z2c6?cid=479)

- What do I do when there are warnings during the build process?
    - Basically, make sure clang command doesn't give warning
    - https://piazza.com/class/ktachtjrz9z2c6?cid=486

- What is CMakeList.txt and what do I do with it?
    - If you want to create additional executables, you can add them like this. But if you don't want to add more files, you don't have to make any modifications.
    - you can check if you have added all the executables by compiling the code and making sure all your executables are running.
    - https://piazza.com/class/ktachtjrz9z2c6?cid=514

- How do I print out the error message?
    - You can use cout
    - https://piazza.com/class/ktachtjrz9z2c6?cid=518
    - https://piazza.com/class/ktachtjrz9z2c6?cid=516

- Do we allow a second "s" command if the first "s" command give rise to an error?
    - Yes (https://piazza.com/class/ktachtjrz9z2c6?cid=518_f1), as long as the VE set are correct