

Data Structure Assignment [7]

General information

- Deadline: **2020/12/24 12:00 noon time.**
- Submit your programming assignment to the Moodle system.
- Submitted file format: student-ID Name.zip (e.g. F12345678_王曉明.zip.) Please name the filename of your submitted compressed file after your **student ID number. Otherwise, 20 points will be deducted.**
- Your submitted files have to be organized in a directory-structured manner as follows. Otherwise, **5 points will be deducted.**
 - | -- F12345678_王曉明
 - | -- F12345678_王曉明.pdf
 - | -- code
 - | -- xxxxx.c
 - | -- xxxxx.c
- Your submitted file must contain **Source Code & Readme file.** (Program description in .pdf format)
- **Late submission will not be accepted.**
- There is a “zero tolerance” for plagiarism. You will receive a score of zero if you get caught plagiarizing.

Course Provisions

1. Program execution environment : Windows (Git Bash)
Command: `./hw.exe < input.txt > output.txt`
2. Programming language: C (standard: C11)
(**Languages other than C are not accepted**)
3. Submitted programming homework must include **source code** in .c data type, and **readme document** in .pdf data type. You are required to address the **(1) result screenshot, (2) program structure, (3) program functions** in readme file. Do not just write the pseudo code or even just copy and paste your code!
4. Homework grading is divided into two parts: 80% for the code and 20% for the readme file. The remaining grading standards are determined by the TAs.

TA office hours of the course:

Wed. 11:30 – 13:00

Lab location: CSIE Bldg. Room 65302

If you have any question, please make an appointment in advance.

You can also mail us about your questions.

TA e-mail: ta_dblab.csie.ncku.edu.tw

Programming homework

Activity On Edge Network

Write a program that allows a user to find all critical paths from a given AOE network.

Input:

The input file, which describes an AOE network, has the following data. The first line of the file contains an integer N ($0 < N < 100$), which indicates the number of activities (i.e., edges) of the network. The succeeding N lines of the input file are the information of the network activities (i.e., edges). Each of these N lines consists of four integers. They are respectively the activity a_i , the starting vertex v_j of a_i , the ending vertex v_k of a_i , and the weight w of this activity, in which i, j, k , and w are all integers and $0 \leq i, j, k, w < 100$. Notice that i, j, k , and w in the INPUT file are each separated by ONE empty space. For example, the 5th line of the INPUT file gives “3 1 4 1”, which means that the activity a_3 's starting vertex is v_1 , its ending vertex is v_4 , and its weight is 1.

Output:

The output file consists of two parts.

- (a) Activity early-late time: The first part gives each activity's index i , the earliest time $e(i)$, and the latest time $l(i)$, each separated by an empty space. For example, the 9th line of the OUTPUT file, which is 8 7 10, means that the activity a_8 's earliest time is 7 and the latest time is 10. Notice that in this part the activities have to be printed in ASCENDING ORDER of the activity index.
- (b) Critical paths: Print out the activity indices of all the critical paths in the last line. Notice that the activity indices printed in this line must be in ASCENDING ORDER and each separated by ONE empty space. In this particular example (referring to the following graph), the critical paths contain activities $a_0, a_3, a_7, a_6, a_{10}, a_9$. However, you need to print them in i 's ascending order (of a_i). Hence, the printout shown in the OUTPUT (referring to the last line) is 0 3 6 7 9 10 (which indicate the critical activities $a_0, a_3, a_6, a_7, a_9, a_{10}$, respectively)

Note:

- Assume that the activity index and event index are consecutive, and the input will be in ascending order of activity index.
- Just use **scanf()** to get the input content and **printf()** to print out the results.
- The format of the output is required to be the same as in the example. Your answer will be considered incorrect if any redundant spaces or extra lines exist after the last output string.

Execution:

Your homework must use the following command to execute in the **Windows Git Bash terminal**, and your output file format must be the same as that in the given output file. In addition to the given input data files, there will be another two hidden input data files used to grade your homework.

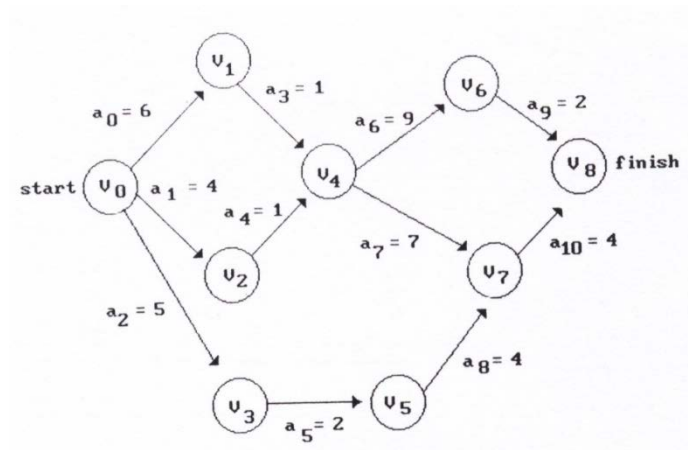
Step 1. `gcc -std=c11 ./*.c -o hw7`

Step 2. `./hw7.exe < input0_windows.txt > ans_output0_windows.txt`

Step 3. `diff ./output0_windows.txt ./ans_output0_windows.txt`

Input

1	11
2	0 0 1 6
3	1 0 2 4
4	2 0 3 5
5	3 1 4 1
6	4 2 4 1
7	5 3 5 2
8	6 4 6 9
9	7 4 7 7
10	8 5 7 4
11	9 6 8 2
12	10 7 8 4



Output

1	0 0 0
2	1 0 2
3	2 0 3
4	3 6 6
5	4 4 6
6	5 5 8
7	6 7 7
8	7 7 7
9	8 7 10
10	9 16 16
11	10 14 14
12	0 3 6 7 9 10