Data Structure Assignment 9

Programming Homework 1

Activity On Edge (AOE) Network

Write a C/C++ program that allows the user to input an AOE network. Assume activity i is represented by edge <*u, v*>and its duration is a_i. The program should calculate and output the the following:

(a) a table of all events with their *earliest* and *latest times*,

(b) a table of all activities with their *early* and *late times*, the *slack*, and the *criticality*,

(c) the critical network.

Input

The first line contains an integer number N ($0 < N < 100$) indicating the number of activities. The following N lines are the information of activities. Each of these N lines consists of four integer numbers, representing the activity index i ($0 < i < 100$), the start vertex of activity u, the end vertex of activity v ($0 \leq u,v < 100$), and the duration of activity a_i ($0 < a_i < 100$) respectively.

Output

The output consists of three parts. Each part is separated by an empty line.

(a) Event table: Print out each event's index *j*, earliest time *ee[j]*, and latest time *le[j]*.

(b) Activity table: Print out each activity's index *i*, early time *e(i)*, late time *l(i)*, slack *l(i)-e(i)*, and criticality status (if *l(i)-e(i)*==0： Yes; else： No).

(c) Critical network: Print out the index of the activities in the critical network. It does not need to be in order.

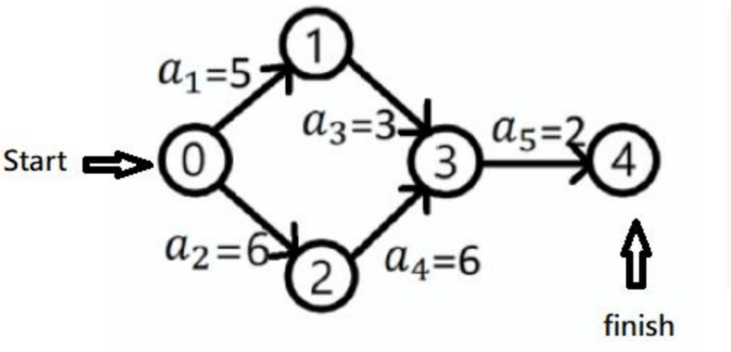
Note:

- Assume that the activity index and event index are consecutive, and the input will be in ascending order of activity index.
- Please use formatted output. You may follow the format as shown in the example output. Points will be deducted if your output's readability is too low.

Please name your submitted file as "aoe.c"/ "aoe.cpp" for this homework.

Example1: Input

```
5
1 0 1 5
2 0 2 6
3 1 3 3
4 2 3 6
5 3 4 2
```



Example1: Output

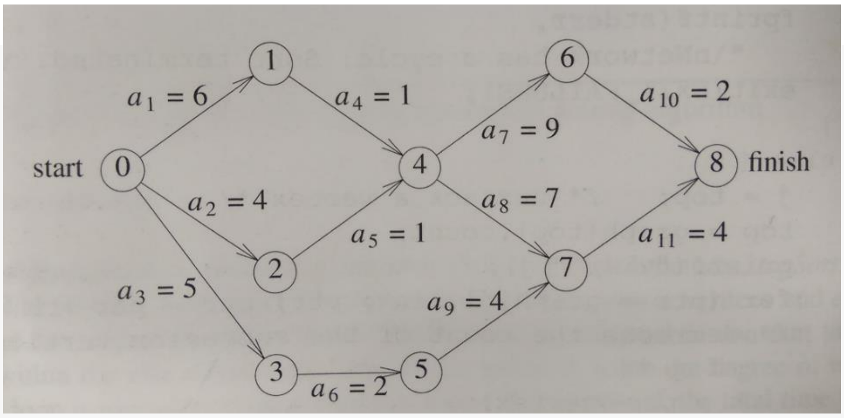
```
event    ee       le
0        0        0
1        5        9
2        6        6
3        12       12
4        14       14

act.     e        l        slack    critical
1        0        4        4        No
2        0        0        0        Yes
3        5        9        4        No
4        6        6        0        Yes
5        12       12       0        Yes

2 4 5
```

Example 2: Input

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



Example 2: Output

```
event   ee      le
0       0       0
1       6       6
2       4       6
3       5       8
4       7       7
5       7       10
6       16      16
7       14      14
8       18      18

act.    e       l       slack   critical
1       0       0       0       Yes
2       0       2       2       No
3       0       3       3       No
4       6       6       0       Yes
5       4       6       2       No
6       5       8       3       No
7       7       7       0       Yes
8       7       7       0       Yes
9       7       10      3       No
10      16      16      0       Yes
11      14      14      0       Yes

1 4 7 8 10 11
```

Programming Homework 2 (Optional)

K-Shortest Paths (KSP)

Efficient management of networks requires that the shortest route from one point (node) to another is known; this is termed the shortest path. It is often necessary to be able to determine alternative routes through the network, in case any part of the shortest path is damaged or busy. The k-shortest paths represent an ordered list of the alternative routes available. [1]

Find **the top '*k*' shortest paths** from the source node to the destination node on the network.

Input

Input a **directed** graph that represents a network. Assume that there is at most one uni-directional link between each node and there is no self-loop.

The first row of input consists of an integer N ($2 \leq N \leq 100$) that indicates the number of nodes on the network. Besides, the nodes are indexed from 1 to N. ***Node 1*** **represents the source node and *node N* represents the destination node of the network.**

The second row of input consists of an integer M that indicates the number of links on the network.

Then, each of the following M rows consists of 3 integers that describes a link. The first 2 integers, A and B, indicate that the link consists of A and B and A points to B. The last integer represents the traffic (weight) of the link.

The last row of input consists of an integer k that indicates the number of shortest paths to be found.

Output

There will be k rows that indicate the k shortest paths being found. They should be printed out in descending order according to their cost. (That is, print out the k paths from the shortest one to the least short one.) Each row consists of several integers that indicate the nodes along the shortest path, through the source to the destination.

Hint: ***Yen's algorithm*** is one of the well-known KSP algorithms. Refer to this paper [1] for more details. You can adopt ***Yen's algorithm*** if you have no idea for this homework.

Note:

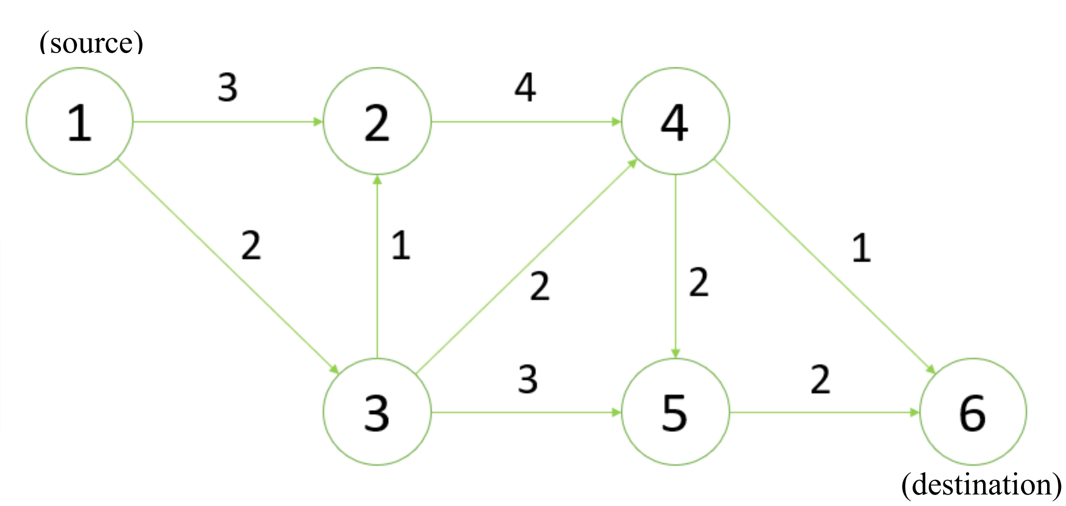
If you are going to use another existing KSP algorithm other than ***Yen's algorithm***, you have to include the citation and write down how it works in your readme file.

You are also encouraged to propose your own KSP algorithm. In your readme file, you should describe your proposed algorithm **as detailed as** that in **this paper** [1].

Please name your submitted file as "ksp.c"/ "ksp.cpp" for this homework.

Input

```
6
9
1 2 3
1 3 2
2 4 4
3 2 1
3 4 2
3 5 3
4 5 2
4 6 1
5 6 2
3
```



Output

```
1 3 4 6
1 3 5 6
1 2 4 6
```

Reference

[1] Brander A.W., Sinclair M.C. (1996) A Comparative Study of k-Shortest Path Algorithms. In: Merabti M., Carew M., Ball F. (eds) Performance Engineering of Computer and Telecommunications Systems. Springer, London, pp. 370-379.

General information:

- Deadline: **2020/1/1 23:55**
- Submit your programming assignment to Moodle system.
- Submitted file format: student-ID_Name.zip, e.g. F12345678_王曉明.zip
- Your submitted file must contain Source Code & Readme file (Program description)
- Late homework will not be accepted
- There is a “zero tolerance” for plagiarism. You will receive a score of zero if you get caught plagiarizing.

資料結構課程規定

1. 程式執行環境: Windows、Linux。

2. 程式語言: C/C++

3. 程式作業只需提供 source code 和 readme 的說明文件。Source code 只接受.cpp 和.c 檔，其餘檔案類型恕不接受，說明文件請含括您的程式內容的解說，例如，程式執行流程，程式架構，如何設計功能等，請不要複製題目或複製程式碼註解貼上。

4. 紙本作業請列出推論過程，僅列出答案而未列出過程者，不給分。繳交紙本作業時，記得寫名字以及學號，請務必記得用 A4 紙張作答，若超過一張，請自行裝訂起來再交給助教，否則將斟酌扣分。

5. 紙本作業與程式作業請勿抄襲，如有發現一律 0 分計算。

6. 程式作業上傳至 moodle 各章節底下的繳交區。

7. 程式作業與手寫作業皆不接受遲交與補交。程式作業在公布之後的兩個禮拜內將會開放上傳繳交，請同學們盡早完成作業，避免在最後期限內的一、兩個小時上傳 moodle 導致發生問題。

8. 每個程式的程式分數佔 80%，說明文件佔 20%，若並未完成作業題目所有要求、 所交程式碼無法執行或執行結果錯誤，將會依照題目要求和執行結果的完程度評分，其餘的評分項目由當次作業批改的助教來決定。

9. 每次作業壓縮檔(例如 zip, 7z 檔)名稱必須以學號命名，若未註明一律扣該次成績 20 分。

本課程的 TA 時段如下:
星期二（Tues.）p.m. 3:00 - 5:00
星期四（Thu.）p.m. 6:00 - 8:00

若是同學有資料結構相關問題可至高速網路實驗室(資訊系新大樓 5F,65503 室)詢問。為免同學撲空，如同學無法在助教時間前來，煩請事先與助教預約時間。如有任何問題請寫信給助教。

助教 何允中 e-mail: p76071323@mail.ncku.edu.tw

潘文傑 e-mail: p76081433@mail.ncku.edu.tw

蘇奕瑋 e-mail: p76081108@mail.ncku.edu.tw

張傑閔 e-mail: p76084499@mail.ncku.edu.tw

蘇雨菲 e-mail: p76075068@gs.ncku.edu.tw

Course Provisions

1. Program execution environment : Windows、Linux

2. Programming language : C/C++ (Languages other than C/C++ are not accepted)

3. Submitted programming homework must include source code in .cpp or .c data type, and readme document. You are required to address the **program architecture**, **program functions and how you design your program** in readme file. Do not just write the pseudo code or even just copy and paste your code!

4. You won't get any point for paper homework if you only write the answers without addressing your process and reasons. **Please do your work on A4 papers. If there is more than one page, please staple them together, and write your student id & name on each page**. Points will be deducted otherwise.

5. **There is a "zero tolerance" for plagiarism. You will receive a score of zero if you get caught plagiarizing.**

6. Please submit your programing homework to moodle.

7. Late homework is not accepted.

8. Programming homework grade is divided into two parts: 80% for the code and 20% for the readme file. **Partial points will still be awarded if the output results of your program are partly correct**. The remaining grading standards are decided by the TAs.

9. **Please name the filename of your submitted compressed file (e.g. zip, 7z) after your** student ID number. 20 points will be deducted otherwise.

TA time of the course:

Tues. 15:00 - 17:00

Thu. 18:00 - 20:00

If you have any question, please come to our lab at TA time (CSIE Bldg. Room 65503). If you are not available to come at TA time, please make another appointment with the TAs. You can also mail us about your questions.

TA	何允中 e-mail: p76071323@mail.ncku.edu.tw
	潘文傑 e-mail: p76081433@mail.ncku.edu.tw
	蘇奕瑋 e-mail: p76081108@mail.ncku.edu.tw
	張傑閔 e-mail: p76084499@mail.ncku.edu.tw
	蘇雨菲 e-mail: p76075068@gs.ncku.edu.tw