Assignment #4

430.329-002 Introduction to Algorithms
Bohyung Han

Due 23:59 5 December, 2020

Note

- T.A. contacts
  - Sanghyeok Chu: sanghyeok.chu@snu.ac.kr
  - DongHwan Jang: jh01120@snu.ac.kr

- If you have any questions about the assignments, please upload them to eTL.

- **You must complete the assignments by yourself.** Otherwise, there will be a severe penalty in your grade.

- Assignments must be completed and submitted by 23:59 as a zip file to the eTL assignment board. Late submission is not allowed.

- You should submit one zip file which contains one pdf (for document) and one python(for code) file. **Please write your name and student ID on the top right of the document.**

- Please title your files as mentioned below: **All cases that do not strictly follow the format will be deducted <u>10%</u> from the total score.**

  - [zip] [StudentID]_assignment4.zip
  - {e.g.} 2020-12345_assignment4.zip

  - [code] [StudentID]_assignment4-1.py
  - {e.g.} 2020-12345_assignment4-1.py

  - [document] [StudentID]_assignment4-2.pdf (Document)
  - {e.g.} 2020-12345_assignment4-2.pdf

1 [Code] Find the shortest path [100 pts]

There are n bus stops connected by bus routes. Each route is charged with a bus fare(> 0). When all bus route information is given in the form of a list, we want to find the path that travels from the source to the destination **through a maximum of k bus stops** (including source and destination) **at the lowest price**. For the given source and destination, please output the cheapest total fare, and -1 if there is no such path.

The skeleton code(2020-00000_assignment4-1.py) and the example input(input.txt) for the assignment will be provided. Please implement your function in that file. **Note that only the python standard library(https://python.readthedocs.io/en/stable/library/index.html) and numpy are available.**

Input:

- The first line is n (the number of bus stops), the second line is the [src, dst, bus fare] list, the third line is the source, the fourth line is the destination, and the fifth line is k (the number limit of bus stops in a path).

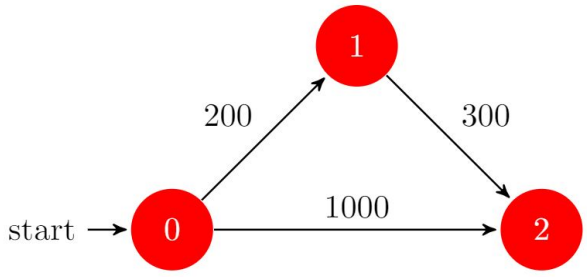
Note:

- Nodes are numbered from 0 to $n-1$. n is at most 100. k is from 2 to n.
- There is only one edge connecting source u and destination v.
- The source and destination cannot be the same.

Example 1) :
Input:
3
[[0,1,200],[1,2,300],[0,2,1000]]
0
2
3
Output: 500

Example 2) :
Input:
3
[[0,1,200],[1,2,300],[0,2,1000]]
0
2
2
Output: 1000

Explanation:



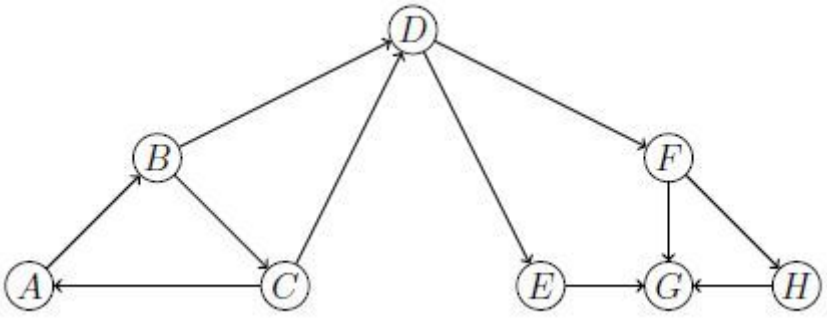
If $k = 3$, we can travel through $0 \rightarrow 1 \rightarrow 2 = 500$

If $k = 2$, we have to go $0 \rightarrow 2$ directly, which costs 1000.

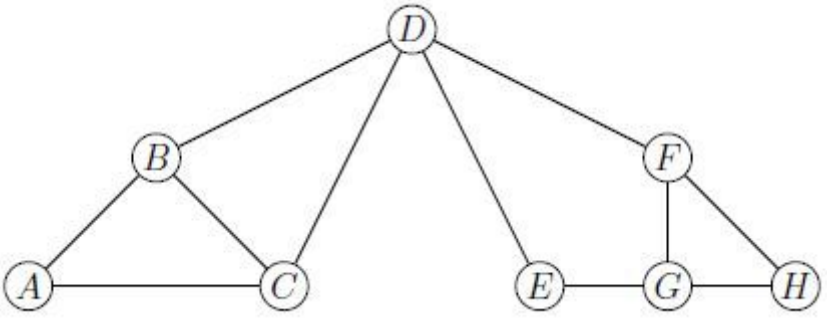
2 [Document] Solve the following problems [100 pts]

1. **[30 points]** To classify whether a graph has cycles or not, we are going to label every edge through executing the DFS on the graph. Label every edge as **T** if it's a tree edge, **B** if it's a back edge, **F** if it's a forward edge and **C** if it's a cross edge. Answer the following questions. Weights of the edges are the same. If the vertex has more than one outgoing edge, process them in alphabetical order.

(a) **[10 points]** Run DFS on the graph below, starting from vertex A and label all the edges as one of **T, B, F and C**.

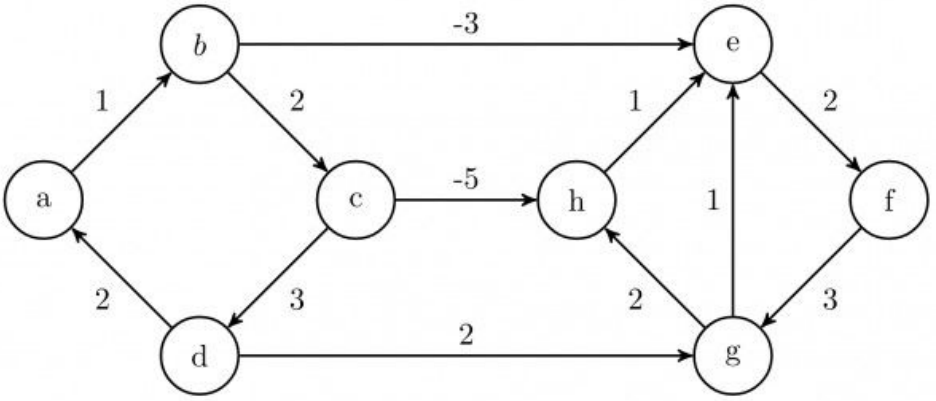


(b) **[10 points]** Run DFS on the graph below, starting from vertex A and label all the edges as one of **T, B.**



(c) **[10 points]** Based on the observations in (a) and (b), describe the way how we can detect cycles by using DFS.

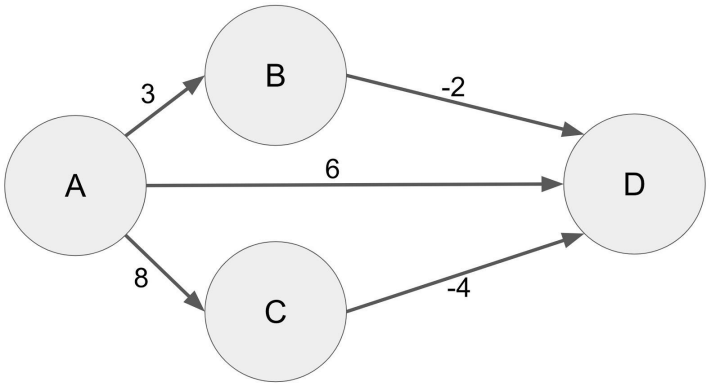
2. **[20 points]** Run the Bellman-Ford algorithm on the below graph, using vertex **a** as the source.



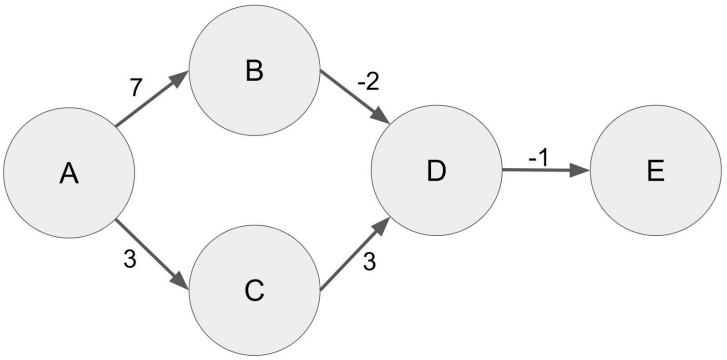
3. **[30 points]** As discussed in the class, the Dijkstra's algorithm works only for the graph with positive weights. For the following two graphs, we will check what would happen if the graph has negative weights. Run the Dijkstra's algorithm on the given two graphs, using vertex **A** as the source, with the following conditions. You need to show all your work.

- If the algorithm produces the correct lowest-cost path to every other node, then say "correct".
- Else if the algorithm produces the wrong path, for all such nodes, write the path the algorithm produces and write the correct lowest-cost path.

(a)



(b)



4. **[20 points]** Suppose there is a weighted, directed graph G that some of its edges have negative weights. To find the single-source shortest path, you add a large constant to every edge of G so that the weights of the new graph G' become positive. Do the shortest paths from Dijkstra(G', source) are still the shortest paths in G? If you think this is true, give a proof. Otherwise, provide a counterexample.