50.004 Introduction to Algorithms, Fall 2018
Singapore University of Technology and Design — 6 Nov, 2018
Gemma Roig, Ioannis Panageas, Subhajit Datta — Home Exercises - Problem Set 2

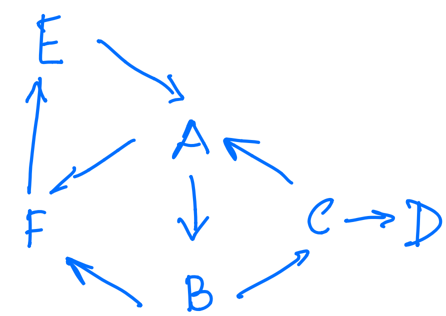
Student Information

Name: Wang Tiandu Student ID: 1002963
Due Date: 29 Nov 23:59.
Submit answers on eDimension in pdf format. Submission without student information will **NOT** be marked! Any questions regarding the homework can be directed to the TA through email (contact information on eDimension).

Exercise 1

Given the following set of adjacency list:

1.



$$adj(A) = [B, F]$$
$$adj(B) = [C, F]$$
$$adj(C) = [A, D]$$
$$adj(D) = []$$
$$adj(E) = [A]$$
$$adj(F) = [E]$$

1. Construct its corresponding **directed** graph.

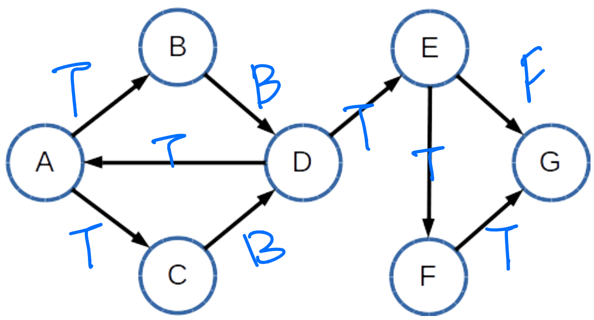
2. Based on the constructed graph, write the order that the nodes are visited **for each type of graph search** when starting at node **B**. Assume that neighbours of the same node are visited in alphabetical order.:

   a. Breath-first Search (BFS) B → C → F → A → D → E

   b. Depth-first Search (DFS) B → C → A → F → E → D

3. For the DFS, perform a topological sort and write the order of nodes in **decreasing** order of finish times, starting from node **B**. Assume that neighbours of the same node are visited in alphabetical order. B, C, D, A, F, E

4. Perform DFS on the following graph, starting at node **D**.
   Label every edge of the graph with 'T' if it is a tree edge, 'B' if it is a back edge, 'F' if it is a forward edge, 'C' if it is a cross edge. Assume that neighbours of the same node are visited in alphabetical order. *Please refer to CLRS pg 609 for the definitions of the edges, under the "Classification of edges" heading if unsure.*

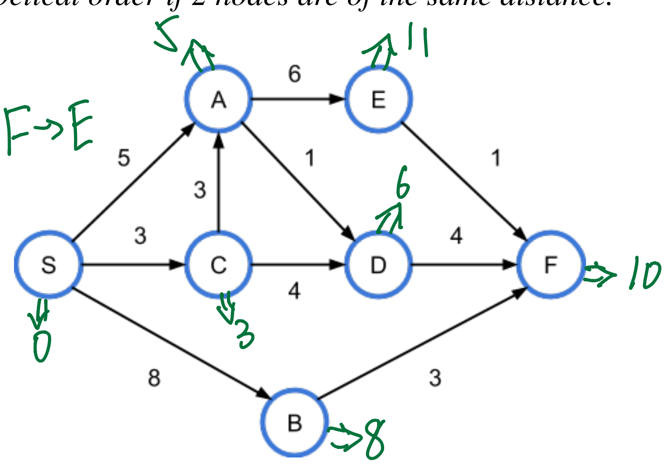


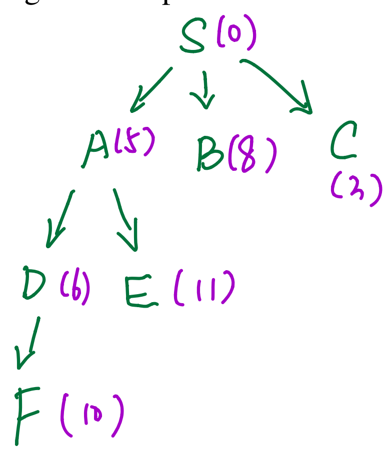
Exercise 2

1. Run Dijkstra's algorithm on the graph below, starting from vertex S. What is the order in which vertices get removed from the priority queue? What is the resulting shortest-path tree? *Follow alphabetical order if 2 nodes are of the same distance.*

Order :

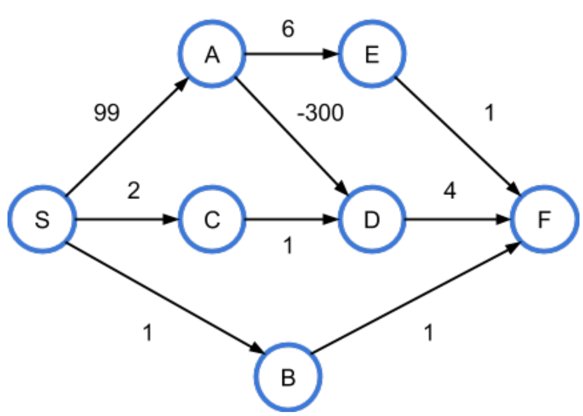
S→C→A→D→B→F→E





2. Now run Dijkstra's algorithm on graph below, starting from vertex S. What is the order in which vertices get removed from the priority queue? If unable to run Dijkstra's algorithm, please explain why not.

Dijkstra's algorithm cannot be applied on this graph because the weight of edge (A, D) is negative. Since in Dijkstra's algorithm, once the vertex is removed from the priority queue, it is assumed to contain the shortest path and it will never be changed again. In this case, when vertex A is visited, vertex D is already moved from priority queue and the shortest path of D cannot be updated



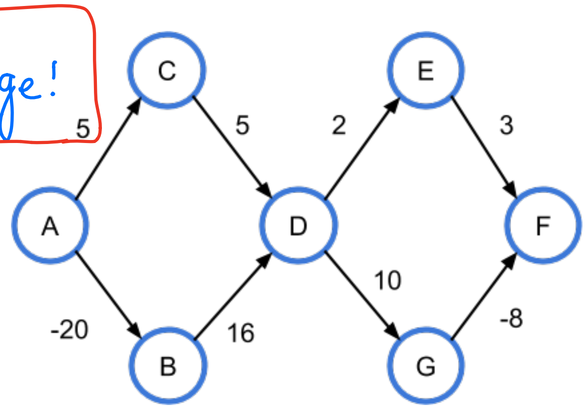
Exercise 3

1. Under what condition does Bellman-Ford algorithm fail to produce the correct shortest path?

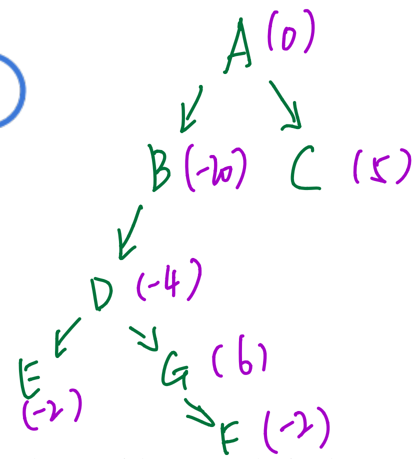
   When there is negative cycle present in the cycle.

2. Use the Bellman-Ford algorithm on the graph below. Assuming that the starting node is A, show the iteration table and the resulting shortest path tree to **all the other nodes**. Please also provide the order of relaxations of the edges. .

See in the last Page!



resulting shortest Path tree



Exercise 4

You would like to buy some groceries far from where you are staying and to avoid a wasted trip due to lack of money, you would want to carry as much money as you can before leaving. Since you would be travelling far, you want to limit your carrying weight to a **maximum of 7g**. Assuming that only coins of value **$1, $5 and $10** are available with their corresponding weights as **1g, 2g and 5g** respectively, you would like to maximise the amount of money you carry while not carrying above your limit. Table 1 illustrates the value of the coins and their corresponding weights.

Weight	1g	2g	5g
Value	$1	$5	$10

Table 1: Coins' weight and value

1. Please solve the problem using the **Greedy approach**. Explain your solution and comment whether it is optimal and why.

   Last Page!

2. Please solve the problem using the **Dynamic programming approach**. Write down the DP equation. (You can answer with a table, pseudocode, text description, or any format that can clearly present your idea.)

   Last Page!

Exercise 3.

	time=0	1	2	3	4	5	6
$\delta(A)$	0	0	0	0	0	0	0
$\delta(B)$	∞	-20	-20	-20	-20	-20	-20
$\delta(C)$	∞	5	5	5	5	5	5
$\delta(D)$	∞	-4	-4	-4	-4	-4	-4
$\delta(E)$	∞	-2	-2	-2	-2	-2	-2
$\delta(F)$	∞	-2	-2	-2	-2	-2	-2
$\delta(G)$	∞	6	6	6	6	6	6

Order: $A \to B$, $A \to C$, $B \to D$,
$C \to D$, $D \to E$, $D \to G$
$E \to F$, $G \to F$

Exercise 4.

1. For \$1 coin: $\frac{\$1}{1g} = \$1/g$

   For \$5 coin: $\frac{\$5}{2g} = \$2.5/g$

   For \$10 coin: $\frac{\$10}{5g} = \$2/g$

So in greedy algorithm, we choose \$5 coin as first preference.
As a result

$$7g = \underbrace{2g}_{\$5} + \underbrace{2g}_{\$5} + \underbrace{2g}_{\$5} + \underbrace{1g}_{\$1}$$

Total money: $5+5+5+1 = \$16$

It is optimal.

Since Dynamic Programming always gives optimal solution, and result of greedy algorithm is as same as the result of Dynamic Programming. Thus, the result of greedy algorithm is an optimal solution.

(2)

	weight / coin	j=0	j=1	j=2	j=3	j=4	j=5	j=6	j=7
i=0	\$0	0	0	0	0	0	0	0	0
i=1	\$1	0	1	2	3	4	5	6	7
i=2	\$5	0	1	5	6	10	11	15	16
i=3	\$10	0	1	5	6	10	11	15	16

The dynamic Programming function

$$\underbrace{tab[i]}_{\text{row}}\underbrace{[j]}_{\text{column}} = max(tab[i-1][j],\ tab[i][j-weigh[i]] + Value(i))$$

$$\{i, j \mid 0 < i \leq \underbrace{3}_{\text{\# of different kinds of coins}},\ 0 < j \leq \underbrace{7}_{\text{Max volume of storage}} \}$$

In this case, the initial case is

from tab[0][0] to tab[0][7] are all 0

And tab[1][0], tab[2][0], tab[3][0] are all 0