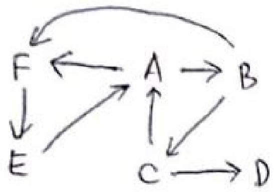
Home Exercises - Problem Set #2

Nov 2018

Ashlyn Goh 1002840 C102

Exercise 1

Q1



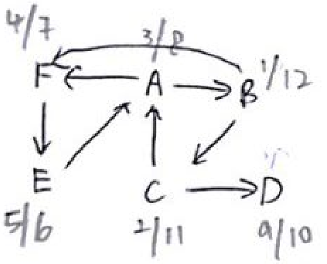
Q2 (a) Breadth first search

B, C, F, A, D, E

(b) Depth first search

B, C, A, F, E, D

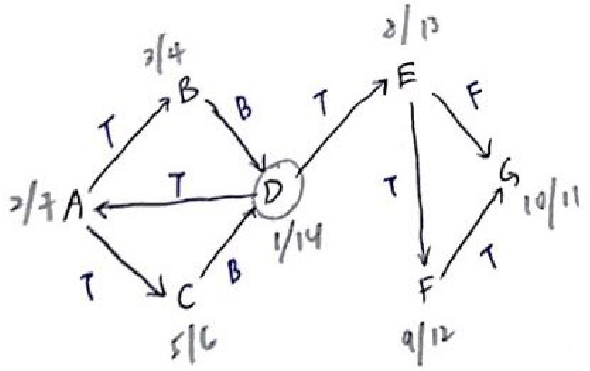
Q3



Decreasing order of finishing times:

B, C, D, A, F, E

Q4

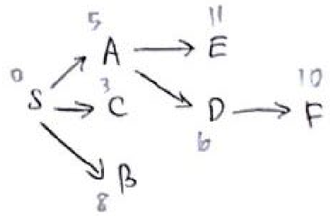


Exercise 2

Q1

Order: S, C, A, D, B, F, E

Shortest path tree:



Q2 In this case, we would not get an optimal solution if we run Dijkstra's algorithm and thus is "unable to run" the algorithm. This is because there are negative weights in this graph. Dijkstra's algorithm for the shortest path relies on the fact that after we visit a vertex, we would not ever find a shorter path to it. Hence, it requires non-negative weights to work. In this case with negative weights, we should not run Dijkstra's algorithm.

Exercise3

Q1 Bellman-Ford algorithm fails when there exists a negative cycle in the graph because when there is a negative cycle, one can keep traversing it, reducing the cost of the path. Thus, there exists no finite shortest path to some vertices when a negative cycle exists.

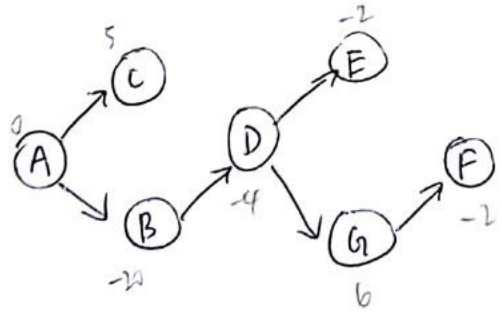
Q2 Order of relaxations

G→F, E→F, D→G, D→E, C→D, B→D, A→C A→B

Iteration Table:

Iteration	A	B	C	D	E	F	G
1	0	-20	5	∞	∞	∞	∞
2	0	-20	5	-4	∞	∞	∞
3	0	-20	5	-4	-2	∞	6
4	0	-20	5	-4	-2	-2	6
5	0	-20	5	-4	-2	-2	6
6	0	-20	5	-4	-2	-2	6

shortest path tree:



Exercise 4

Q1

In the Greedy approach, we can solve the problem via taking as many of the coin with the highest value-to-weight ratio, coin A. Thereafter, when the bag cannot accommodate an additional coin A, we can take the next coin with the highest value-to-weight ratio if the bag can still accommodate the weight of that coin.

We can solve the problem through the following steps:

Take one $5 coin	weight left = 5g
Take another $5 coin	weight left = 3g
Take another $5 coin	weight left = 1g
Take a $1 coin	weight left = 0g

Weight	1g	2g	5g
Value	$1	$5	$10
Value/Weight	1	2.5 (highest)	2

Maximum amount of money carried = $5(3) + $1 = $16

This solution appears to be optimal since I cannot find another combination of coins carried (with the weight restriction applied) that has a greater total value than $16. In this case, there is an unlimited supply of each coin and the lightest coin is 1g, hence the greedy approach works in this case. One is able to fill the bag to capacity in this question since the lightest coin is 1g. Without any empty space in the bag, an optimal solution can be obtained using the Greedy approach for this question.

Q2

In this question, we are allowed to have multiple copies of each item. Hence, the DP equation would be slightly different from the general 0-1 Knapsack question (where multiple copies are not allowed).

Let DP[i, X] be the best value one can get using only item from 1 to i placed in a bag of capacity X.
Let DP[i, 0] = 0 and DP[0, X] = 0

Overall DP Equation: $\boldsymbol{DP[i, X] = max\{DP[i-1, X], v_i + DP[i, X - s_i]\}}$

Solving using tabular method:

		Capacity of Bag (X)						
Using item 1 to i		X = 1	X = 2	X = 3	X = 4	X = 5	X = 6	X = 7
	i = 1	1	2	3	4	5	6	7
	i = 2	1	5	6	10	11	15	16
	i = 3	1	5	6	10	11	15	16

Hence, maximum amount of money carried = $16