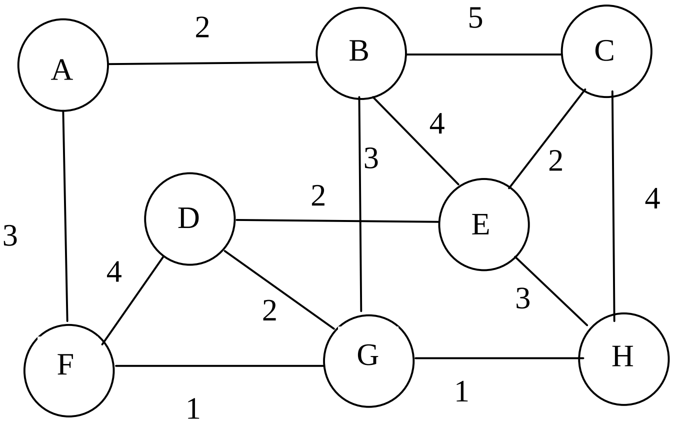
CS575 Assignment 3 (summer 2020)
Due on July 14 (by 11:59pm)

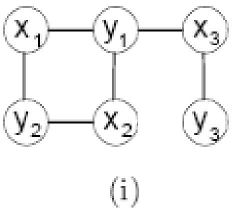
Topics: Depth-first search, Breadth-first search, topological sorting, and linear programming, greedy algorithm

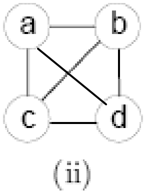
[Part A] Theory [72%]:

1. (12%)
(1) (6%) Find the minimum spanning tree using the Prim's algorithm for the following graph starting at node A. Show the order that the nodes are added into the minimum spanning tree. Draw the minimum spanning tree and indicate the weight summation of the MST.
(2) (6%) Redo the above (1) except using the Kruskal's algorithm

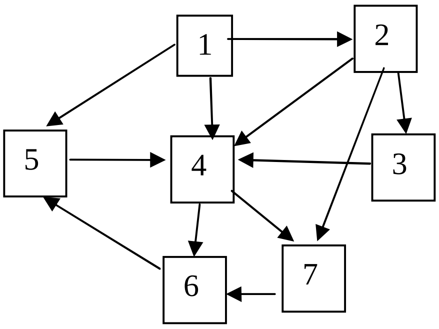


2. (a) [6%] Design an algorithm which can automatically indicate whether or not a graph *G* contain a cycle. Using the following two graphs as two examples to indicate what data structure to represent (i) and (ii), respectively.

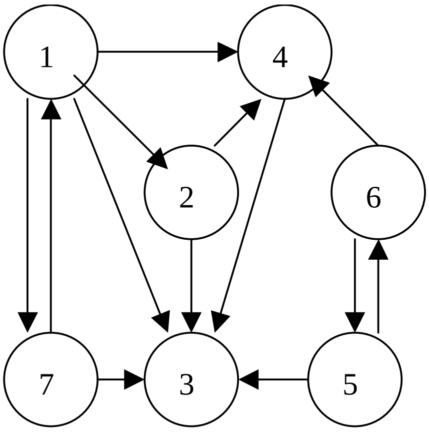




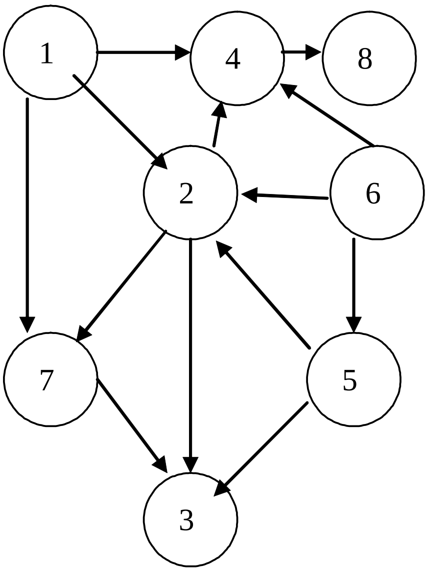
(b) [6%] Give the order that the nodes will be visited when doing a BFS starting at the node 1 of the following graph:



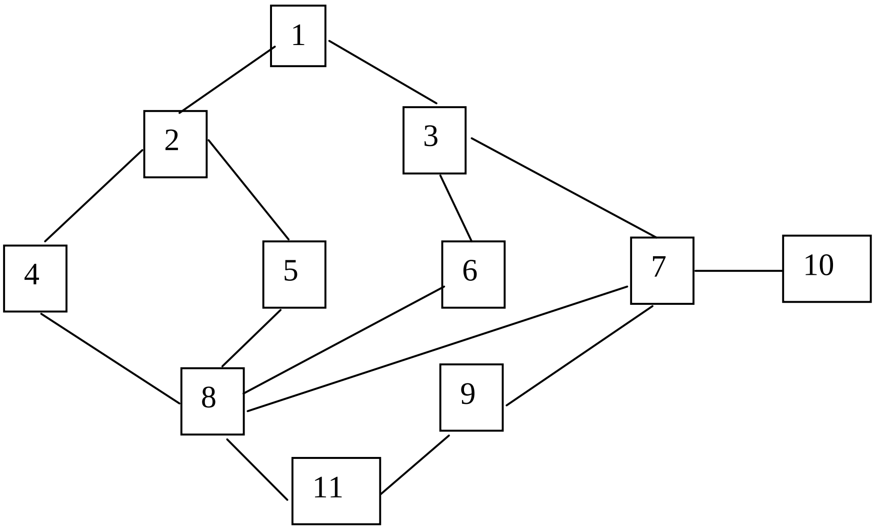
3. [8%]Apply depth first search to the following graph. Show the graph after a new edge has been identified as a tree, forward, or cross edge, back edge (indicate one of the four types). Show also the current color of all the nodes. (NOTE: assume starting from node 1, and go to node 2 first).



4. [8%] Apply topological sort to the graph below. Show the sequence of the nodes found by your application. Include also the discovery and finishing time of each node. (Assume the starting node is 1, the second node to go is 7. Also node 2 will be selected before node 4.)



5. [8%] A bipartite graph G(V, E) is an undirected graph whose vertices can be partitioned into two disjoint sets V1 and V2 = V-V1 with the properties that no two vertices i nV1 are adjacent in G and no two vertices in V2 are adjacent in G. All edges go between the two sets V1 and V2. Is the following graph G a bipartite graph? Write your algorithm to determine whether the graph G is bipartite and the two disjoint sets V1 and V2 if it is a bipartite.

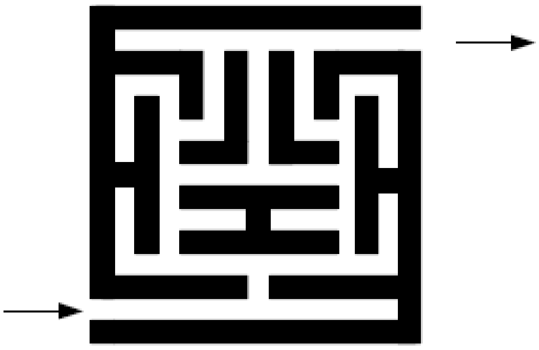


6. [8%]

One can model a maze by having a vertex for a starting point, a finishing point, dead ends, and all the points in the maze where more than one path can be taken, and then connecting the vertices according to the paths in the maze.

a. Construct such a graph for the following maze.

b. Which traversal–DFS or BFS–would you use if you found yourself in a maze and why?



7. [8%] A merchant plans to manufacture two models of home computers at costs of $250 and $400, respectively. To sell the computers, the $250 model yields a profit of $45 and the $400 model yields a profit of $50. The merchant estimates that the total monthly demand will not exceed 250 units. Using Graphical Method, find the number of units of each model that should be stocked in order to maximize profit. Assume that the merchant does not want to invest more than $70,000 in computer inventory.

8. [8%] You are downloading *n* files (songs, programs) from the web to a minimum number of CDs. Each CD can store exactly B bytes. More than one file can be stored on each CD, but a single file cannot be stored on more than one CD (no split). Each file *i* has a size s_i bytes ($s_i \leq$B). Describe a greedy algorithm to solve this problem. It may not get optimal solution, If so, give a counter example.

[Part B]: Programming Part (28%)

B.1 Graph Algorithm (14%)

Suppose a CS curriculum consists of n courses, all of them mandatory. The prerequisite graph G has a node for each course, and an edge from course v to course w if and only if v is a prerequisite for w. Find and algorithm that works directly with this graph representation, and computes the minimum number of semesters necessary to complete the curriculum (Assume that a student can take any number of courses in one semester). The running time of your algorithm should be linear.

Using following example to justify your answer:
The CS Department requires fifteen one-semester courses with the prerequisites shown below:

cs1
cs2
cs3
cs4 requires cs2
cs5 requires cs4
cs6 requires cs1 and cs3
cs7 requires cs4
cs8 requires cs5 and cs6
cs9 requires cs7
cs10 requires cs9
cs11 requires cs8
cs12 requires cs3
cs13 requires cs6
cs14 requires cs4 and cs6
cs15 requires cs14

Your task is to determine the minimum number of semesters needed to finish the degree.

(Hint: Represent the courses and their prerequisites as a DAG. Finding the minimum number of semesters translates to a simple graph problem, e.g., Using adjacency-matrix representation in BFS).

Please provide:

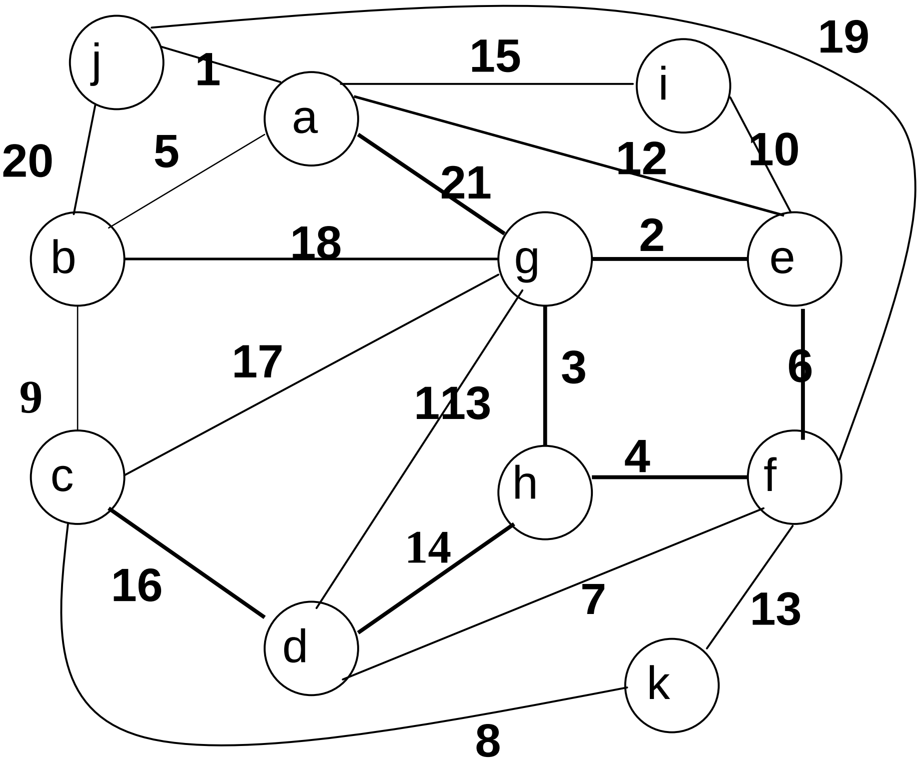
1. Manually plot the DAG (3%)
2. Explain the algorithm that you are going to implement by the Pseudo-code, and indicate the minimum number of semesters necessary to finish the degree. (3%)
3. Write and run your program to print out the result for verification (i.e., minimum number of semesters) (8%)

B.2 Shortest path algorithm design and programming implementation (14%)

(1) (6%)

Suppose there are a group of cities, which are modeled by a strongly connected undirected graph G = (V, E) with positive edge weights representing the distances between two cities. A particular city is the capital city a $\in$ V. Give an efficient algorithm for finding shortest paths between all pairs of cities, with the one restriction that these paths must all pass through the capital city. (Note that: we allow one city to be passed more than once if needed).

(2) (8%) Programming: Implement your designed algorithm, and print out the shortest path from node *d* to node *i* via node *a* using the graph shown below. Suppose the capital city is node “*a*”.


j
1
a
15
i
19
20
5
12
10
21
b
18
g
2
e
17
3
6
9
113
c
h
4
f
14
16
7
13
d
k
8