CS 101 Fall 2021 - Quiz 8-B

11/30/2021 - 20 Minutes **Name:** **ID number:**

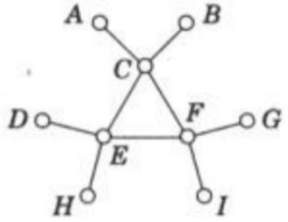
The following questions are choice questions, each question may have **one** or **multiple** correct answers. Select all the correct answer, you will get half points if you choose a strict subset(excluding empty set) of the right answer.

*Note: You should write those answers **in the box** below.*

Question 1	Question 2	Question 3
C	B	AB

Question 1(4pts):

How many spanning trees dose the following graph have?



(A) 1

(B) 2

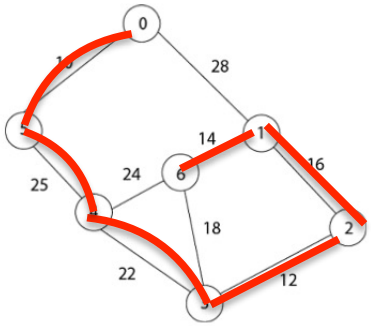
(C) 3

(D) 4

(E) None of the above.

Question 2(4pts):

In the figure below, using Kruskal's algorithm to compute the MST, which edge should we choose last?



(A) (0,1)

(B) (4,5)

(C) (3,6)

(D) (4,6)

(E) None of the above.

Question 3(4pts).

Which of the following algorithms reflect(s) the idea of greedy algorithms?

(A) Kruskal

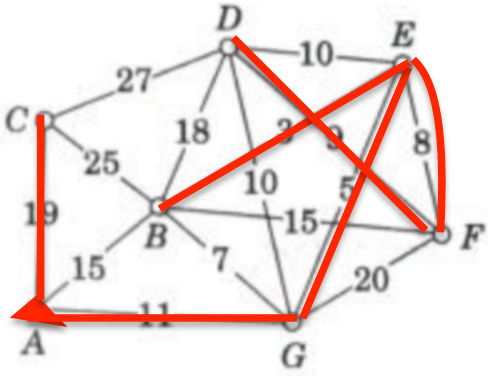
(B) Prim

(C) Quicksort

(D) Mergesort

Question 4(4pts):

Write down the sequence of edges added to the minimum spanning tree using Prim's algorithm. Suppose we start from vertex "A". (You can randomly choose one edge if you meet two edges with the same weight.)

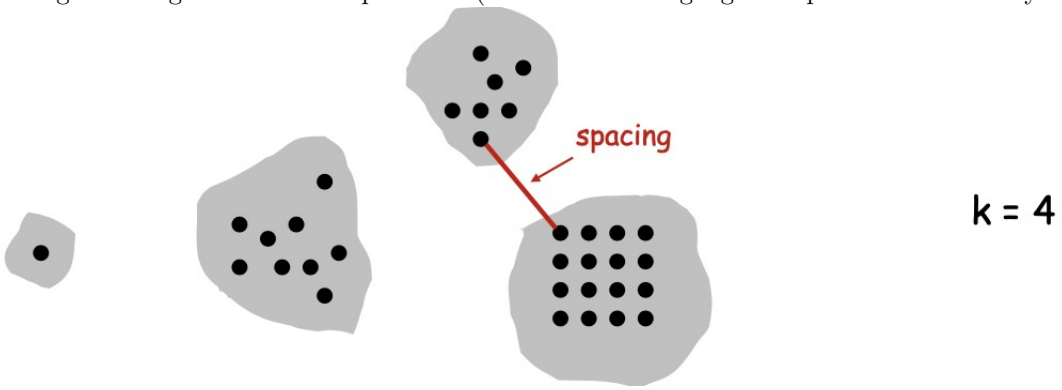


Answer: AG GE EB EF FD AC

Question 4(6pts):

Suppose we are seeking to divide the objects in U into k groups, for a given parameter k. We say that a k-clustering of U is a partition of U into **k nonempty** sets $C_1, C_2, ..., C_k$. We define the spacing of a k-clustering to be the minimum distance between any pair of points lying in different clusters. Given that we want points in different clusters to be far apart from one another, please seek the k-clustering with the maximum possible spacing.

Please give an algorithm for this problem. (Both natural language and psedo-code are okay to show your answer)



Single-link k-clustering algorithm.

Create n clusters, one for each object.

Find the closest pair of objects such that each object is in a different cluster; add an edge between them and merge the two clusters.

Repeat n-k times until there are exactly k clusters.

Key observation. This procedure is precisely Kruskal's algorithm (except we stop when there are k connected components).

Remark. Equivalent to finding an MST and deleting the k-1 most expensive edges.