EECS 233 Written Assignment #4

Due December 7, 2018 before 11:59 pm (EST)
100 pts

Submission Guidelines

Submit all your solutions in the form of a PDF file to Canvas. You can either **type them up** or **write on papers and scan/take picture of them**. If you choose the latter, please make sure that your hand writing is **clear and readable**.
Please name your file as **W4_YourCaseID_YourLastName.pdf**

1) Consider a **connected** undirected graph with N nodes.
   a) If this graph has the **smallest** possible number of edges,
      i) How many edges does it have with respect to N? **(4 pts)**
      ii) Draw an example of such a graph with N = 5 nodes. **(4 pts)**
      iii) What are such graphs called? (Check the lecture notes for terminology) **(4 pts)**

   b) If this graph has the **largest** possible number of edges,
      i) How many edges does it have with respect to N? **(4 pts)**
      ii) Draw an example of such a graph with N = 5 nodes. **(4 pts)**
      iii) What are such graphs called? (Check the lecture notes for terminology) **(4 pts)**

2) Consider the quicksort algorithm.
   a) If the **first** element of the input array is always used as a pivot,
      i) What is the asymptotic runtime (expressed as Big-O) for sorted input? **(4 pts)**
      ii) What is the asymptotic runtime (expressed as Big-O) for reverse sorted input? **(4 pts)**

   b) If the **middle** element of the input array is always used as a pivot,
      i) What is the asymptotic runtime (expressed as Big-O) for sorted input? **(4 pts)**
      ii) What is the asymptotic runtime (expressed as Big-O) for reverse sorted input? **(4 pts)**

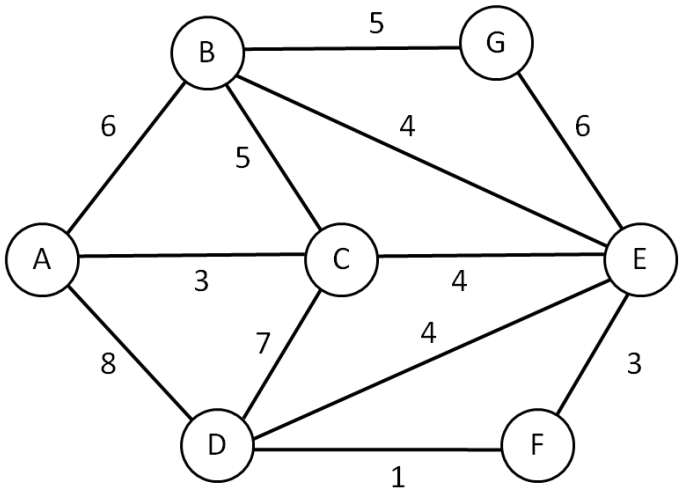
3) Suppose you have an array of N objects (as a reminder, in Java, the array would actually contain references to the objects). Each object has a key field and some other data fields (such as id). The key field is a string that can have only two distinct values: "red" and "blue". Design a time-wise O(N) algorithm to rearrange an array so that all "red" objects precede "blue" objects. **You may use only constant extra space (You cannot use a new array).**
   a) Describe your algorithm (You do not need to write code. Just explain how your algorithm operates in plain terms). **(12 pts)**
   b) Apply your algorithm to the following array step by step: **(12 pts)**

(key, id):	Red, 3	Blue, 5	Blue, 7	Red, 1	Red, 4	Blue, 2	Red, 6

   c) Argue that your algorithm is O(N) time-wise. **(6 pts)**

   Note: a non-acceptable solution would be to just count the number of reds and then rewrite the key field of the first k elements with the string "red" and the rest with "blue". This would change the nature of the objects. For example, if your objects are flowers such that they have a *key* field for "color" and another field *id* for "species" and if you just over-write the color field you would end up with corrupted data (with incorrect color - species pairs).

4) Consider the following undirected graph:



a) Start from node A and apply **Dijkstra's** algorithm to find a spanning tree.
   i) List (in order) which nodes are added to the spanning tree, and draw the resulting tree. **(12 pts)**
   ii) Is the tree found by Dijkstra's algorithm a minimum spanning tree? **(3 pts)**

b) Start from node A and apply **Prim's** algorithm to find a spanning tree.
   i) List (in order) which nodes are added to the spanning tree, and draw the resulting tree. **(12 pts)**
   ii) Is the tree found by Prim's algorithm a minimum spanning tree? **(3 pts)**