CSCB63 Assignment 2- Summer 2021

40 Points

Due Date: 12th July 2021 11:59 pm on MarkUs

Note: Make sure you complete this assignment on your own without help from any resources other than class notes and textbooks. Read the guidelines on Plagiarism. https://utsc.calendar.utoronto.ca/4-academic-integrity

Make sure you are capable of applying BFS, DFS, Prim's algorithm, Kruskal's algorithm and Dijkstra's algorithm to an appropriate type of graph (directed, not directed or weighted).

Q. 1 (5 marks)

Given a maximum heap H, what is the time complexity of finding the 3 largest keys in H?
What is the time complexity of finding the ith largest key in H?
Explain if you can improve time complexity of finding ith largest key in H.

Q. 2 (5 marks)

Determine whether the following claim is true and either prove the claim or it's negation.

> *Given a graph G with n vertices such that for every* $v \in V$ *,* $\boldsymbol{deg}(v) \geq \frac{n}{2}$ *, then G is one connected component.*

Q. 3 (5 marks)

In class, we have (or will soon) seen Dijkstra's algorithm to find all the shortest paths from a single source in a weighted graph. In this question, you will construct a graph that has many shortest paths, in fact, your graph will have 3^n shortest paths between a source vertex s and a sink vertex t, where the number of vertices is a function of the form $cn + k$ for some $c, k \in N$. More precisely, prove:

> *For every natural number* n*, there is an undirected graph of* $cn + k$ *vertices such that for some pair of vertices* s *and* t *in the graph, there are* 3^n *shortest paths from* s *to* t*.*

You can select the constants k and c to make it work.

Q. 4 (5 marks)

Consider an undirected graph $G = (V, E)$ with non-distinct, non-negative edge weights. If the edge weights are not distinct, it is possible to have more than one MST. Suppose we have a spanning tree $T \subset E$ with the guarantee that for every $e \in T, e$ belongs to some minimum-cost spanning tree in G. Can we conclude that T itself must be a minimum-cost spanning tree in G?

Give a proof or a counter example with explanation.

Q. 5 (10 marks)

We can model the build process of an object as a directed graph. For example, suppose we are building a house. The walls can't be painted until the drywall is installed which cannot happen until after the studs are built. Suppose we model the process with a vertex representing the components of the object and a directed edge from a to b if a must be completed before b can be completed. Notice that if this directed graph has a cycle, then there is no way to construct the object.

a) Give an algorithm to determine whether a directed graph has a cycle. What should your complexity be?
b) Using DFS, construct an algorithm that either returns a valid ordering of the vertices to build the object or a cycle confirming no such ordering exists. Again, what should your complexity be?

Q. 6 Programming Question: (10 Marks)

For the coding part of the assignment, you'll be implementing min-heaps using arrays as discussed in class. The relevant files in the starter code for this section are `minheap.h`, `minheap.c` and `test_minheap.c`.

First look at `minheap.h` to see how the data structures are stored for the min-heap implementation and understand what's going on. In particular, we are having to modify our structure to support fast `decrease_priority()` operations. The header file explains what these changes are.

Once you are familiar with the data layout, open `minheap.c` and complete the functions there to manipulate the heap. Be thoughtful about how you write a code, writing some common helper functions can greatly reduce the amount of work you need to do.

Lastly, the test `minheap.c` file provides some basic tests for your implementation. These are not comprehensive, and you should test for correctness yourself by adding more thorough test cases and tracing portions of your algorithm by hand to compare. You will need to use your min-heap implementation for the next section of the assignment, so make sure that your implementation is correct before you begin. You don't want to be debugging multiple different parts of your code when something goes wrong.

Marking Scheme (10 marks)

• 1 marks: `newMinHeap()`
• 3 marks each for: `heapPush()`, `heapExtractMin()`, `heapDecreasePriority()`
• You will receive 0 marks if your code does not compile, or fails all the test cases, with possible penalties for compiling with warnings.

You will need this implementation for your next Assignments' coding question.

Submission Instructions:

Late Assignment **will not be** graded.
Submit following files: Assignment2.pdf
minheap.c
Make sure the names of the files exactly match the names given above.