CSCI 570 Homework 2

Due Date: Sept. 21, 2023 at 11:59 P.M.

1. In the SGM building at USC Viterbi, there is a need to schedule a series of n classes on a day with varying start and end times. Each class is represented by an interval $[start\_time, end\_time]$, where $start\_time$ is the time when the class begins and end_time is when it concludes. Each class requires the exclusive use of a lecture hall.
   (a) To optimize resource allocation, devise an algorithm using binary heap(s) to determine the minimum number of lecture halls needed to accommodate all the classes without any class overlapping in scheduling. (7 points)
   (b) Analyze and state its worst-case time complexity in terms of n. (3 points)
2. The Thomas Lord Department of Computer Science at USC Viterbi is working on a project to compile research papers from various departments and institutes across USC. Each department maintains a sorted list of its own research papers by publication date, and the USC researchers need to combine all these lists to create a comprehensive catalog sorted by publication date. With limited computing resources on hand, they are facing a tight deadline. To address this challenge, they are seeking the fastest algorithm to merge these sorted lists efficiently, taking into account the total number of research papers (m) and the number of departments (n).
   (a) Devise an algorithm using concepts of binary heap(s). (7 points)
   (b) Analyze and state its worst-case time complexity in terms of m and n. (3 points)
3. In an interstellar odyssey, a spaceship embarks on a journey from a celestial origin to a distant target star, equipped with an initial fuel capacity of `'currentFuel'` units. Along the cosmic highway, there are space refueling stations represented as an array of `'spaceStations'`, each defined as `[distanceToStationFromOrigin, fuelCapacity]`. There are `'n'` space stations between the celestial origin and the target star.

The objective is to determine the minimum number of refueling stops required for the spaceship to reach the target star, which is located `'targetDistance'` light-years away. The spaceship consumes one unit of fuel per light-year traveled. Upon encountering a space station, it can refuel completely by transferring all available `'fuelCapacity'` units from the station. The challenge is to calculate the `'refuelStops'` needed for a successful voyage to the target star or return -1 if reaching the destination remains unattainable with the available fuel resources.

(a) Devise an algorithm using concepts of binary heap(s). (7 points)

(b) Analyze and state its worst-case time complexity in terms of n. (3 points)

4. You are tasked with performing operations on binomial min-heaps using a sequence of numbers. Follow the steps below:

   (a) Create a binomial min-heap $H1$ by inserting the following numbers from left to right: 3, 1, 13, 9, 11, 5, 7, 15. (2 points)

   (b) Perform one `deleteMin()` operation on $H1$ to obtain $H2$. (2 points)

   (c) Create another binomial min-heap $H3$ by inserting the following numbers from left to right: 8, 12, 4, 2. (2 points)

   (d) Merge $H2$ and $H3$ to form a new binomial heap, $H4$. (2 points)

   (e) Perform two `deleteMin()` operations on $H4$ to obtain $H5$. (2 points)

   Note: It is optional to show the intermediate steps in your submission. Only the five final binomial heaps ($H1$, $H2$, $H3$, $H4$, and $H5$) will be considered for grading. So, please ensure that you clearly illustrate your final binomial heaps ($H1$, $H2$, $H3$, $H4$, and $H5$). You can use online tools like draw.io for drawing these heaps.

5. If we have a k-th order binomial tree (B_k), which is formed by joining two B_{k-1} trees, then when we remove the root of this k-th order binomial tree, it results in k binomial trees of smaller orders. Prove by mathematical induction. (10 points)

6. Given a weighted undirected graph with all distinct edge costs. Design an algorithm that runs in $O(V + E)$ to determine if a particular edge e

is contained in the minimum spanning tree of the graph. Pseudocode is not required, and you can use common graph algorithms such as DFS, BFS, and Minimum Spanning Tree Algorithms as subroutines without further explanation. You are not required to prove the correctness of your algorithm. (10 points)

7. Given a weighted undirected graph with all distinct edge costs and $E = V + 10$. Design an algorithm that outputs the minimum spanning tree of the graph and runs in $O(V)$. Pseudocode is not required, and you can use common graph algorithms such as DFS, BFS, and Minimum Spanning Tree Algorithms as subroutines without further explanation. You are not required to prove the correctness of your algorithm. (10 points)

8. There are N people with the i-th person's weight being w_i. A boat can carry at most two people under the max weight limit of $M \geq \max_i w_i$. Design a greedy algorithm that finds the minimum number of boats that can carry all N people. Pseudocode is not required, and you can assume the weights are sorted. Use mathematical induction to prove that your algorithm is correct. (10 points)

9. Given $N > 1$ integer arrays with each array having at most M numbers, you are asked to select two numbers from two *distinct* arrays. Your goal is to find the maximum difference between the two selected numbers among all possible choices. Provide an algorithm that finds it in $O(NM)$ time. Pseudocode is not required, and you can use common operations for arrays, such as min and max, without further explanation. Prove that your algorithm is correct. You may find proof by contradiction helpful when proving the correctness. (10 points)

10. There are N cities (city 1, to city N) and some flights between these cities. Specifically, there is a direct flight from every city i to city $2i$ (no direct flight from city $2i$ to city i) and another direct flight from every city i to city $i-1$ (no direct flight from city $i-1$ to city i). Given integers a and b, determine if there exists a sequence of flights starting from city a to city b. If so, find the minimum number of flights required to fly from city a to city b. For example, when $N = 10$, $a = 3$, and $b = 9$, the answer

is 4 and the corresponding flights are $3 \to 6 \to 5 \to 10 \to 9$. You are not required to prove the correctness of your algorithm. (10 points)