Name:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CS 146 Spring 2016

Final Exam

1. / 10

2. / 10

3. / 20

4. / 10

5. / 10

6. / 20

7. / 10

8. / 10

1. Recall Ω is the asymptotic lower bounds for an algorithm.

Also, Recall that the comparison-sorting problem is Ω(n log n). This problem instead considers the problem of finding the minimum value of a collection assuming elements can only be compared (resolving ties for the minimum arbitrarily).

(a) Consider this bad argument for showing that the minimum-finding problem is Ω(n log n): We can find the minimum of n elements by putting all elements in an array arr, sorting the array, and returning the first element). Since the sorting step is Ω(n log n), the minimum-finding problem is Ω(n log n). Explain what is wrong with this argument in 1-2 English sentences. (2 pt)

(b) Give a simple argument that the minimum-finding problem is Ω(n). ( 3 pt)

(c) Can we or can we not conclude from your answers to (a) and (b) that the minimum-finding problem is Ω(n log n)? Explain your answer. (5 pt)

2.



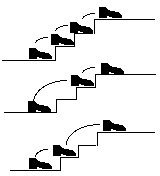






1. Even though the graph has negative weight edges, step through Dijkstra’s algorithm to calculate supposedly shortest paths from A to every other vertex. Show your steps in a table. Cross out old values and write in new ones, from left to right within each cell, as the algorithm proceeds. Also list the vertices in the order which you marked them known. (5 pt)
2. Did Dijkstra’s algorithm find the wrong path to some of the vertices? For just the vertices where the wrong path was computed, indicate both the path that was computed and the correct path. (1 pt)
3. Can a single edge be removed from the graph such that Dijkstra’s algorithm would happen to compute correct answers for all vertices in the remaining graph? If so, which one? (4 pt)

3. You’re climbing a stair case. It takes n steps to reach to the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?



Input : 3

Return : 3

Steps : [1 1 1], [1 2], [2 1]

This picture describes n = 3 above, with a solution of 3.

Provide a dynamic programming solution in pseudocode. No written explanation, but rather using functions, code notation, as explained throughout the semester.

4. Does Prim’s algorithm work when there are negative weights in the graph? Prove why or why not. Either use a counter example to prove it doesn’t work. Or explain why it works.

5. Describe an algorithm to find the minimum number of edges that need to be removed from an undirected graph so that the resulting graph is acyclic. You cannot assume the graph is connected.

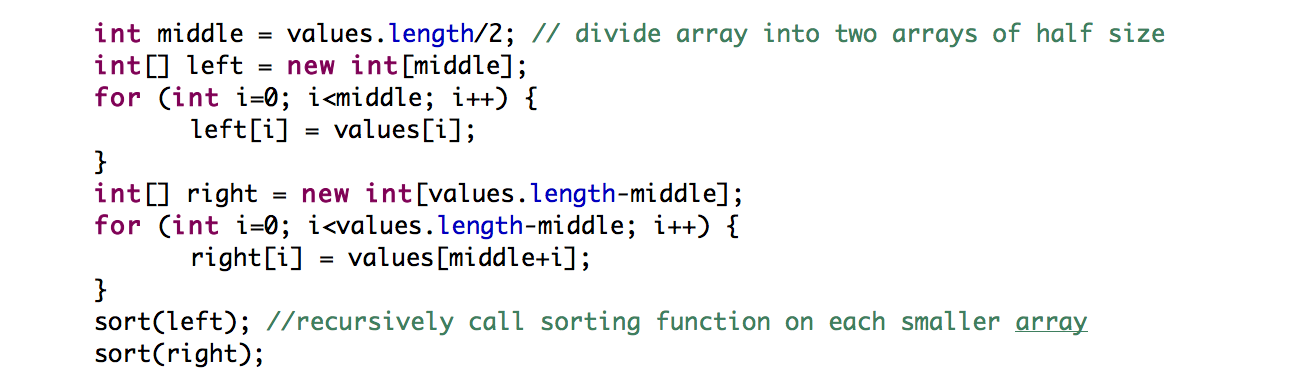
For the problem, you don’t need to use pseudo code. An explanation in English is ok.

6. Mergesort

1. Given the following array as input, illustrate how the Mergesort algorithm performs. To illustrate the Mergesort’s behavior, start with the dividing of the array until the end condition of the recursive function is met and then show how the merge is performed. (10 pt)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 8 | 4 | 11 | 2 | 22 | 7 | 0 |

1. Given the function below, that divides the original array in two arrays of half size, develop the Java code that combines two sorted arrays. Use the result from problem 6a) to develop that code. (7 pt)



C. Answer the following three questions: What is the complexity of dividing and merging an array of length n? What is the complexity of dividing arrays in Mergsort? What is the overall complexity of Mergesort? Give your answers in big O notation. (3 pt)

7. Why is P = NP such an important computer science question? Why is it so interesting?

8. For a linked list, which is a better sort: mergesort or quicksort? Why? For an array, which is a better sort: mergesort or quicksort? Why?

Extra Credit:

How is my (instructor’s) name spelled? First and last name. You’ll get 6 points.