**CS4335 Design and Analysis of Algorithms (Midterm, 2021)**

**If you think some questions are ambiguous, you may write your assumptions clearly. However, you assumption should not trivialize the questions.**

**Question 1. (20 points)**

**(a) (10 points)** For the interval scheduling problem, the set of jobs (si, fi) are as follows:

(0, 2), (1, 5), (2, 6), (3, 4), (5, 7) (8, 12), (7, 10), and (9, 13).

Use a greedy algorithm to compute the maximum number of compatible jobs. You should give main steps. What is the running time of the greedy algorithm?

**(b) (8 points)** For the interval partitioning problem, the set of lectures (si, fi) are as follows:

(0, 2), (0, 4), (2,5), (3, 6), (3, 5), (4, 7), (5, 8) and (6, 9).

Use a greedy algorithm to compute the minimum number of classrooms to accommodate all the lectures. You should give main steps.

**(c) (2 points)** For the interval partitioning problem given in (b), what is the depth of the problem?

**Question 2. (20 points)**

1. **(7 points)** Find the minimum spanning tree for the graph in Figure 1 using Kruskal’s algorithm.

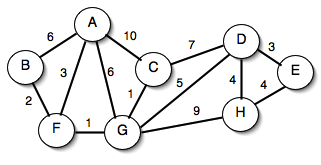


Figure 1

1. **(8 points)** Find the minimum spanning tree for the graph in Figure 1 using Prim’s algorithm.
2. (**5 points**) Is the path between a pair of vertices in a minimum spanning tree of an undirected graph necessarily a shortest path? Justify your answer.

**Question 3. (15 points)**

Use Dijkstra’s algorithm to compute a shortest path from *a* to *i* in the following graph. You should give main steps.

A picture containing watch, different, arranged

Description automatically generated

Figure 2

**Question 4 (15 points)**

1. (**9 points)** For the list:2, 1, 5, 8, 9, 10, 4, 7, 6, 13, 14, and 11. Suppose we have sorted the two halves as list1: 1, 2, 5, 8, 9, 10; and list2: 4, 6, 7, 11, 13, 14. Calculate the number of inversions with one number in list1 and the other number in list2 using O(n) operations. Immediate steps are required.
2. **(3 points).** Assume T(n) is the running time for the following algorithm. List the recursive relation, and with it, what is T(n) in terms of big O notation?

**FindMax(A, k, n)**

**Input:** Array A of size n, and an integer k<n

**Output:** the maximum element from A[k], A[k+1], …, A[n-1]

**if** k<n-1

**return** max(A[k], FindMax(A, k+1, n))

**else return** A[k]

**Initial call FindMax(A, 0, n)**

1. (**1 point**) Suppose *T(1)=1,* and *T(n)=T(n-1)+n*. What is *T(n)* in terms of big O notation?
2. **(2 points)** Suppose *T(1)=1,* and *T(n)=T(n/3)+1*. What is *T(n)* in terms of big O notation?

**Question 5. (15 points)**

Given an array of n ≥ 2 **distinct** integers (i.e., no two integers are the same) sorted in ascending order, say [x(1),...,x(n)], we want to find the absolute minimum *difference between the x(i) and i*. For example, for *x = [-10, 9, 10, 12, 13, 16] ,* the minimum *difference d =*|*x(2)−2| =* |*9−2| = 7*.

**(a)** (**5 points)** Use a linear time algorithm to solve the problem.

**(b)** (**5 points**) Use a divide and conquer approach the solve the problem. The running time should be O(logn).

**(c)** **(5 points)** Set up and solve a recurrence equation for part (b) to estimate the running time of your algorithm. Prove that the running time of your algorithm is O(logn).

**Hint:** The difference will first decrease and then increase.

**Question 6.** **(15 points)**

Suppose we have an array of n positive integers. A contiguous subarray A[i .. j] is called a squared interval if the sum of its entries is a squared number. Design a greedy algorithm to compute the maximum number of squared intervals such that every entry in A will be covered at most once. You can state your algorithm in English or in Pseudo code (**5 points**). What is the running time of algorithm in big-Oh (**5 points**)? Prove that your algorithm is correct **(5 points)**.

4, squared

16, squared

36, squared

25, squared

1 7 7 5 5 3 4 7 6 9 1 13 5

Figure 3