Department of CSE, UIU

CT-1 -- CSE 2217 – Fall21 -- Time 45mins -- Total Marks 20

1 Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size *n*, insertion sort runs in $8n^2$ steps, while merge sort runs in $64n \lg n$ steps. For which values of n does insertion sort beat merge sort? 4

Solution: We wish to determine for which values of n the inequality $8n^2 < 64n \lg(n)$ holds. This happens when $n < 8 \lg(n)$, or when $n \le 43$. In other words, insertion sort runs faster when we're sorting at most 43 items. Otherwise merge sort is faster.

2 Find the worst-case time complexity of the following pseudocode and express in Θ notation. 4

```
for i = 1 to n − 1 do
  min = i
  for j = i + 1 to n do
    // Find the index of the i-th smallest element
    if A[j] < A[min] then
      min = j
    end if
  end for
  Swap A[min] and A[i]
end for
```

Solution: $\Theta(n^2)$

3 Solve the following recurrence relations using Master method: [DIY] 4

i. $T(n) = 2T(n/2) + \log n$
ii. $T(n) = 27T(n/3)+n^3$

4 What does it mean when we say that an algorithm X is asymptotically more efficient than Y? Prove that divide and conquer method will give maximum sum subarray in $O(n\log n)$ time when $n>1$ 1+3

solution: An algorithm X is said to be asymptotically better than Y if X takes smaller time than y for all input sizes n larger than a value n0 where n0 > 0

solution: Maximum crossing subarray takes O(n) time. To recursively call with the left and right subarrays, since there are two subarrays, the total time complexity for this portion is 2T(n/2).

T(n) = 2T(n/2)+O(n)

Using Master's theorem, a=2, b=2; $n^{\log_b a} = n$. Case 2 of Master method gives $T(n)=\Theta(n\log n)$, which can be upper bounded by *O(nlogn)*

[note: you may use recursion tree method as well]

5 Given an array of integers A = {-2, 1, -3, 7, -1, 2, 3, -5, 8}, find the Maximum and Minimum using divide-and-conquer. Show the necessary steps to support your answer. [DIY] 4