CSE:221 Section:1

1st Quiz

Marks:30, Time:30 Minutes

1. Write down the psedo-code of heap-sort algorithm for sorting an array in descending order.

2. Consider the following pseudo-code. Estimate the time complexity using BIG-OH notation. You need to answer the following queries.

|  |
| --- |
| BinarySearch(A[0..n-1], value, low, high) {  // invariants: value > A[i] for all i < low  value < A[i] for all i > high  if (high < low)  return not\_found // value would be inserted at index "low"  mid = (low + high) / 2  if (A[mid] > value)  return BinarySearch(A, value, low, mid-1)  else if (A[mid] < value)  return BinarySearch(A, value, mid+1, high)  else  return mid  } |

(i) What is the recurrence relation in terms of input size **n**?

(ii) Solve the above relation using **Recurrence-Tree** method.

(ii) Mention the number of nodes at the bottom level, no. of levels and height of the tree with respect to **n**.

3. Compare Quick-Sort and Insertion-Sort in terms of following properties.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Strategy | | | Time-Complexity | Space Complexity | Stability | In/Out Place |
| Quick -Sort | Divide & Conquer Approach | | | TQ(n)= |  |  |  |
| Divide | Conquer | Combine | TQ(n)= |
|  |  |  | TQ(n)= |
| Insertion-Sort | Incremental Approach | | | TI(n)= |  |  |  |
|  | | | TI(n)= |
|  | | | TI(n)= |