CSE 3131: ALGORITHM DESIGN 1

ASSIGNMENT 2:

Submission due date: 24/11/2022

- *Assignment scores/markings depend on neatness and clarity.*
- *Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should ALWAYS prove the correctness of your algorithms either directly or by referring to a proof in the book.*
- *The marking would be out of 100.*
- *You are allowed to use only those concepts which are covered in the lecture class till date.*
- *Plagiarized assignments will be given a zero mark.*

CO2: to understand various types and aspects of ***sorting and searching algorithms****.*

Sorting and Searching:

Heap and Heap sort:

Sl.No.	*Question*	*PO*	*Level*
1.	Given an array $A[1..n]$ that is first decreasing up to some places from 1 to p (value is unknown) and then increasing from $(p+1)$ to n. More precisely, there is a coordinate $1 \leq p \leq n$ such that for all $i < p$, $A[i] > A[i + 1]$, and for all $i \geq p$, $A[i] < A[i + 1]$. Your goal is to find the smallest element in this array. Design an efficient algorithm to find the smallest element in this array that has the same asymptotic running time as Binary search. Explain the operation using a suitable example.	PO2, PO3	L2, L3, L4
2.	The INSERTIONSORT algorithm operates by picking a *key* element from the unsorted part of the array and inserting it in its appropriate position in the sorted subarray. i. Sort the given array A = <28, 52, 17, 35, 24, 48, 11, 20, 17, 30> using Insertion sort. Show each step of the operation. ii. Assuming an equal probability for the *key* element to be inserted at each of the available positions, derive the average case time complexity for INSERTIONSORT algorithm. Compute the best and worst case time complexity from the above formulation explaining the corresponding input instances. iii. Justify the online nature of the INSERTIONSORT algorithm with an example.	PO1, PO2	L2, L3, L4
3.	Suppose we are comparing the implementations of Insertion sort and Heap sort on the same machine. For the input of size n, Insertion sort runs in $8n^2$ steps, while Heap sort run in $64n\log n$ steps. For which values of n, Insertion sort outperforms Heap sort?	PO1, PO2	L2, L3
4.	In a binary max-heap containing 100 elements, if the root is at position one, i. Find the position of the parent of a node i that is present at position 19 and check if i is a left or right child of its parent. ii. What will be the height of the node i?	PO1, PO2	L2, L3

<table>
<tr><td></td><td>iii. Find the number of nodes with two children, one child and zero child respectively.
iv. How many nodes are present in the left subtree and right subtree of the root node?</td><td></td><td></td></tr>
<tr><td>5.</td><td>A binary heap contains 100 elements. Compute the following:
i. The position of the first leaf node
ii. Total number of leaves
iii. Height of the heap
iv. Number of nodes at height 3</td><td>PO1, PO2</td><td>L2, L3</td></tr>
<tr><td>6.</td><td>Derive a formula to compute the height of the k^{th} largest element in a binary max-heap of height <i>h</i>, if all the <i>n</i> elements are unique and the array representation of the heap is already sorted. Use the derived formula to compute the height of the 35th largest element in a max-heap containing the elements as the natural numbers from 60 to 1 in sorted order.</td><td>PO1, PO2</td><td>L2, L3</td></tr>
<tr><td>7.</td><td>Convert the given array <i>A</i> = <27, 17,3,16,13,10,1,5,7,12,4,8,9,0> into a:
i. Max-heap
ii. Min-heap
Find the total number of comparisons and calls to MAXHEAPIFY procedure in both the cases.</td><td>PO1, PO2</td><td>L2, L3, L4</td></tr>
<tr><td>8.</td><td>Sort the given array <i>A</i> = <26, 20, 24, 8, 17, 5, 41, 8, 35, 20> in ascending order using Heap sort and find the total number of calls to the MINHEAPIFY procedure.</td><td>PO1, PO2</td><td>L2, L3</td></tr>
<tr><td>9.</td><td>The following array <i>A</i> contains the distinct positive integers from 1 to 13.<table><tr><td>13</td><td>8</td><td>a</td><td>b</td><td>c</td><td>d</td><td>5</td><td>6</td><td>4</td><td>1</td><td>2</td><td>10</td><td>9</td></tr></table>What will be the values of <i>a</i>, <i>b</i>, <i>c</i> and <i>d</i> so that it is a max-heap?</td><td>PO1, PO2</td><td>L2, L3</td></tr>
<tr><td>10.</td><td>You are given a min-heap <i>H</i> with <i>n</i> elements and a search key <i>k</i> (not necessarily present in <i>H</i>). Your task is to print all the keys in <i>H</i> that are less than or equal to <i>k</i> in sorted order.
i. Propose an efficient algorithm to solve this problem.
ii. Supply a tight bound on the running time of your algorithm.</td><td>PO1, PO2, PO3</td><td>L2, L3, L4</td></tr>
<tr><td>11.</td><td>Define a max-heap and its contiguous representation in an array.
i. Convert the array <i>A</i> = <20, 11, 25, 7, 23, 17, 9> into a max-heap
ii. Explain how you insert a key <b>22</b> in the above max-heap</td><td>PO1, PO2</td><td>L2, L3</td></tr>
<tr><td>12.</td><td>Does Heap sort follow the stable sorting property? Justify with an example.</td><td>PO2</td><td>L2, L3</td></tr>
</table>

13.	Write a pseudocode to check whether a given set of data elements is representing a max-heap or not. Then check the given array with values <23, 17, 14, 6, 13, 10, 1, 5, 7, 12> for a max-heap? If not, then build the max-heap from the given array.	PO1, PO2	L2, L3, L4
14.	Illustrate the operation of MAX-HEAPIFY(A, 3) on the array A = <27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0>. Illustrate the operation of HEAPSORT on the array A.	PO1, PO2	L2, L3
15.	What is the running time of HEAPSORT on an array of length n that is already sorted in increasing order? What will be the running time if the array elements are sorted in descending order?	PO1, PO2	L2, L4
16.	Suppose instead of the binary heap, we implement the Priority queue operations using a k-ary ($k \geq 2$) heap. i. How can the elements of the k-ary heap be arranged implicitly in an array? Specifically, where are the children of a node present at position i of the array located? ii. Analyse the priority queue operations (INSERT, EXTRACTMIN) implemented in a k-ary heap using the standard algorithms described for binary heap. What are the number of comparisons performed in the worst case for these operations? iii. Discuss about the optimal value of k to get the best performance on the number of comparisons used in these operations. iv. Suppose we are only interested in minimizing the number of moves (swaps) performed in the priority queue operations. What value of k would you choose?	PO1, PO2	L2, L3, L4

Submission and Grading:

Submit the hard copy of your assignment by the due date, i.e. 24.11.2022.

Part of your assignment grade comes from its "external correctness." This is based on correct output on various sample inputs.

The rest of your assignment's score comes from "internal correctness." Internal correctness includes:
1. Use of methods to minimize the number of steps.
2. Appropriate use of rules, axioms, and suitable diagrams to enhance readability of your responses.

Send a zip folder (name of the zip folder must be your registration number_AD1) containing the code and output file/screen-shot of each program implementation mentioned to the official email id of your AD1 class teacher. On the top of each program, you must mention your full name, registration number, title of the program and date.