AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF COMPUTING, BENGALURU
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
V SEMESTER CSE (July - December 2024)

19CSE302 – DESIGN AND ANALYSIS OF ALGORITHMS

COURSE PLAN (L-T-P-C: 3-0-3-4)

Lab Questions:

Sorting

1. Given an array A that is k-nearly sorted. You need to sort the array A of size N with the best time complexity possible where k<<<N.

a) Which of the basic sorting techniques namely bubble sort, selection sort and insertion sort will be best suited for sorting A as per your understanding. Explain and justify using detailed complexity analysis for all three techniques.

b) In continuation to part (A), implement the best chosen sorting algorithm and any one of the remaining two. Also, plot the graphs for complexity analysis of the techniques implemented on a common data.

2. Insertion sort divides the array into two part, a sorted part and an unsorted part. In each iteration it selects the top element E from the unsorted part, linearly searches for its best position P in the sorted part and inserts the element E at the position P by shifting the array elements in the forward direction

a) Discuss how can we improvise the traditional insertion sort algorithm for its worst case where the data is placed at random in the array with the help of detailed complexity analysis for traditional and improved insertion sort.

b) Implement the traditional and the improved versions of insertion sort and plot graphs for complexity analysis and comparisons.

c) Discuss the impact of the improved insertion sort on the best case and average case with complexity analysis. It is helpful in such cases.? You can choose to plot graphs and check.

3. Given ‘m’ sorted lists/ arrays, each containing ‘n’ elements, print them efficiently in sorted order. For Example:

```
Input: 5 sorted lists of fixed size 4

[10, 20, 30, 40]
[15, 25, 35, 45]
[27, 29, 37, 48]
[32, 33, 39, 50]
[16, 18, 22, 28]

Output:

[10, 15, 16, 18, 20, 22, 25, 27, 28, 29, 30, 32, 33, 35, 37, 39,
40, 45, 48, 50]
```

4. Given an array of size N, find the K largest elements in the array where K<<<N.
For Example:

Input: *[1, 23, 12, 9, 30, 2, 50], K = 3*

Output: *50, 30, 23*

5. Given a set of activities, along with the starting and finishing time of each activity, find the maximum number of activities performed by a single person assuming that a person can only work on a single activity at a time.

6. Write a program to implement inversion count.
Given the Input (non-negative integers): A = { 10 , 1 , 2 , 4 , 13 , 9 , 5 }

The number of inversions that are possible as follows:
{ (10 , 1) , (10 , 2) , (10 , 4) , (10 , 9) , (10 , 5) , (13 , 9) , (13 , 5) , (9 , 5) }
Total count of inversions are: 8

7. Write a program to implement introsort (Introspective sort).