In-Lab Practice Sheet 1

CSE204L - Design and Analysis of Algorithms

January 6, 2023

1. **Design related problems:**

   (a) Translate the Insertion-sort algorithm discussed in the class into a program which takes n numbers (real or integers). Inputs for the program is n and the n numbers.

   (b) Given a sequence of n numbers (real or integers) and a number k (k is one among the n numbers), write an algorithm and the corresponding code to compute the position of k if the given n numbers are arranged in an increasing order, using insertion sort. If the 2, -1, 3, 0, 7 and (k=)3 are the input , your program should output 4 since 3 will be in the fourth position (starting from 1), in the sorted (increasing) order.

   (c) All the alphabets(lower case) of english language a, b, c, ..., y, z are assigned values 1, 2, 3, ..., 25, 26. Given a sequence of n symbols from english alphabet (only lower case), write an insertion-sort based algorithm to arrange the given n symbols, in an increasing order of their values.

   (d) Given a sequence of n numbers (real or integers), write an algorithm (insertion) and the corresponding code to arrange the given n numbers are arranged in such a way that all the negative numbers (if any) are arranged in a descending order and all the positive numbers are arranged in an increasing order with zero (if it is in the input) appearing between the smallest negative number and the smallest positive number. If 7, 3, 2, 4 the output should be 2, 3, 4, 7. If -7, -3, 2, 4 the output should be -3, -7, 2, 4 should be the output. If 7, 3, -1, 0, 2, 4 the output should be -1, 0, 3, 4, 7.

   (e) Given n points P_1,P_2 , ...,P_n with the coordinates (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) respectively, write an insertion-sort based algorithm and the corresponding code to arrange the points in an increasing order of the distance of the point from the origin (with (0, 0) as the coordinates. Distance between any two points (a_1, b_1) and (a_2, b_2) is $\sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$. Input for the code is n and the coordinates of the n points entered with x-coordinate first and then the y-coordinate.

   (f) The insertion sort algorithm, which was discussed in the class is based on the following constraint: The comparison operation will be starting from the rightmost element in the sorted sub-part

while finding the correct position in sorted sub-part. Your task is to write an algorithm and implement the same based following constraint: The comparison operation will be starting from the leftmost element in the sorted sub-part while finding the correct position in sorted sub-part for placing new element.

(g) Write and implement the insertion sort algorithm using recursive approach.

2. **Analysis related problems**

   (a) Execute the insertion-sort algorithm many time with different inputs and Justify the statement " **Best-case running time of $T(n)$ is a linear function of n**".

   (b) Execute the insertion-sort algorithm many time with different inputs and Justify the statement " **Worst-case running time of $T(n)$ is a quadratic function of n**".

   (c) Execute the insertion-sort algorithm many time with different inputs and Justify the statement " **Average-case running time of $T(n)$ is a quadratic function of n**".

   (d) Compute the running time of program (P), t(P) in seconds for the insertion-sort algorithm for different inputs and draw the graph n Vs t(p).

3. **Real Tim Case Study: Hackerearth Problem Monk and Nice Strings**

   (a) Monk's best friend Micro's birthday is coming up. Micro likes Nice Strings very much, so Monk decided to gift him one. Monk is having N nice strings, so he'll choose one from those. But before he selects one, he need to know the Niceness value of all of those. Strings are arranged in an array A, and the Niceness value of string at position i is defined as the number of strings having position less than i which are lexicographicaly smaller than $A[i]$. Since nowadays, Monk is very busy with the Code Monk Series, he asked for your help. **Note**: Array's index starts from 1.
   Input:First line consists of a single integer denoting N.
   N lines follow each containing a string made of lower case English alphabets.

Output:Print N lines, each containing an integer, where the integer in i^{th} line denotes Niceness value of string $A[i]$.
Sample Input: 4 a c d b
Sample Output: 0 1 2 1
Refer this URL: Link