Lab 3.1

3.1. Write a program to sort a given set of elements using the insertion sort. Additionally, determine the time required (in terms of steps) to sort the elements. (Note: assume cost of any basic operation is 1, i.e., $c_1 = c_2 = \ldots = c_8 = 1$).

1) Repeat the experiment for different values of n = 500, 1000, 5000, 10000

2) For each of aforementioned case, consider arrays as sorted, random, and reverse-sorted. Provide the complexity in terms of step count

Note: No keyboard input. Use Random number generator.

n	Sorted	Random	Reverse Sorted
500			
1000			
5000			
10000			

Help: Insertion Sort

```
INSERTION-SORT($A$)                                          cost     # times

1: for $j = 2$ to $n$                                        $c_1$    $n$
2:     $key = A[j]$                                          $c_2$    $n - 1$
3:     // Insert $A[j]$ to the sorted                        $0$      $n - 1$
           sequence $A[1..j - 1]$
4:     $i = j - 1$                                           $c_4$    $n - 1$
5:     while $i > 0$ and $A[i] > key$                        $c_5$    $\sum_{j=2}^{n} t_j$
6:         $A[i + 1] = A[i]$                                 $c_6$    $\sum_{j=2}^{n} (t_j - 1)$
7:         $i = i - 1$                                       $c_7$    $\sum_{j=2}^{n} (t_j - 1)$
8:     $A[i + 1] = key$                                      $c_8$    $n - 1$
```

Help: Random Number Generator

Random Number Generation

```
#include <stdlib.h>
#include <time.h>

srand(time(NULL)); //once

rand()%30; //everytime
```

Generating in sorted order

```
arr[0] = rand()%100;

//for sorted order
for(int i = 1; i < arr_size; i++){
    arr[i] = arr[i - 1] + rand()%30;
}
```

Lab 3.2

3.2. Write a program to compute the nth Magic number (recursively) defined as below and find its time complexity (in terms of number of recursions).

nth magic number MN(n) = MN(n-1) + MN(n-2), whereas MN(0) = 0, and MN(1) = 1

Lab 3.2

3.2. Write a program to compute the nth Magic number (recursively) defined as below and find its time complexity (in terms of number of recursions).

nth magic number MN(n) = MN(n-1) + MN(n-2), whereas MN(0) = 0, and MN(1) = 1

[Divide and conquer approach]

Homework

HW 3.1. Write an program for counting inversions in an array. Inversion is a pair such that for an array A = {a1, a2, a3,...., an}, and ai > aj and i < j.

HW 3.2. Write a program to implement GCD (greatest common divisor) using the following three algorithms.
a) Euclid’s algorithm
b) Consecutive integer checking algorithm.
c) Middle school procedure which makes use of common prime factors.
Study the time complexity. Present some results to show which is more effective.

HW 3.3 Write a program to implement binary search on an array which may has two subsequences: first consists of numbers in ascending order and second in descending order.

Whiteboard

Whiteboard

Whiteboard