

Indian Statistical Institute
CS1
First Year Students
7Jan2022
Closed Book Exam – Duration 3 hours.

Total Marks: 80

ANSWER Question 1, 2,3,4 and Either Q5 OR Q6 (but not both)

Q1. [Total Marks: 2+2+2+2+4+2+2+2+2=20]

For Q1a,b,c,d, e, consider the swap function:

```
void swap(int *x, int *y)
{ int temp = *x;
*x = *y;
*y = temp;
}
```

Suppose we have an array of three integers `int x[] = {20,10,30}`. Are the following statements True (T) or False (F) as applied to the above array? Only T/F answers, no calculation, needed.

Q1a.) `swap(x[1], x[2])` swaps 10 and 30.

Q1b.) `swap (x, x+2)` swaps 10 and 30

Q1c.) `swap(&x[2] , &x[0])` swaps 10 and 20.

Q1d.) `swap(x, &x[1])` swaps 20 and 10.

Q1e.) Using the swap function given above fill in the missing line as indicated in the comment in the function definition given below that swaps `a[i]` with `a[j]`.

```
void swap_i_j(int *a, int i, int j) //swaps a[i] with a[j]
{ //FILL IN THIS MISSING STATEMENT:ONE LINE ONLY }
```

Evaluate the following (remember `%` has higher precedence than `*` / which have higher precedence than `-+`) Only answers, no calculation, needed.

Q1f.) Value of `x` in `x = 3 + 4 % 5 - 6`;

Q1g.) Value of `x` in `x = -3 * 4 % - 6 / 5`;

Q1h.) If initially `x = 1`, then the value of `y` and `x` after evaluation of `y = x++ - 1`;

Q1i.) If initially `x = 1`, then the value of `y` and `x` after evaluation of `y = ++x - 1`,

Q2.[Total Marks: 10]

Complete the following C program by adding appropriate functions and statements as mentioned in the comment in the main program below. Your answer must be ready to be compiled and run.

```
#include <stdio.h>

typedef struct{
char name[30];
int marks;
} st_marks;

main ( )
{
st_marks MarkSheet[5] =
{ {"Samir",20}, {"Jalal",40}, {"John",30}, {"Vidya",60}, {"Ayesha",25} };

/* WRITE THE REST OF THE PROGRAM TO print out the students' names
in ascending order of marks */

}

// You may want to use a modified version of the function given next.

void insertion_sort(int *arr, int length) //sorts an array of int of given length
{ int i, j, tmp;
for(i = 0; i<length; i++) {
for(j = i - 1 ; j >=0; j--){
if( *(arr + j) > *(arr + j +1) )
{
tmp=*(arr + j);
*(arr + j)=*(arr + j + 1);
*(arr + j+ 1)=tmp;
}
else break;
} } }
```

Q3.[Total Marks: 5+10=15]

A linked list of integers is declared as follows:

```
struct Node{  
    int num;  
    struct node *next;  
};
```

Node* top;

Q3a.) Define a function “length” that accepts pointer “top” as parameter and returns the number of nodes in the list.

Q3b.) Define a C function appendNode that CREATES AND appends a new node (with data, integer in this case) at the end of a linked list.

Q4. [Total Marks: 4+6+6+4=20]

Q4a.) Derive the recursion relation for the computational complexity $T(n)$ in the finding the largest integer in an array of n integers by using the fact $\text{Max}(a,b,c, \dots) = \text{Max}(a, \text{Max}(b,c, \dots))$, and using recursion. What is the base case?

Q4b.) Solve the recursion equation in Q5a (of finding maximum) and give asymptotic estimate of $T(n)$ for large n . In other words what is $T(n)$ in the big O notation?

Q4c.)

The Fibonacci sequence is defined as

$F(0) = 1, F(1) = 1, \text{ for } n > 1, F(n) = F(n-1) + F(n-2).$

A recursive algorithm for calculating $F(n)$ is

```
Algorithm Fib( $n$ ) {  
    if ( $n < 2$ ) return 1  
    else return Fib( $n-1$ ) + Fib( $n-2$ )  
}
```

Write the recurrence relation for the time complexity $T(n)$ for this algorithm. Show that $T(n) = O(2^n)$

Q4d.) Another way to calculate the Fibonacci numbers is through a C function such as

```
int fib(int n)
{
    /* Declare an array to store Fibonacci numbers. */
    int f[n + 1];
    int i;

    f[0] = 0;
    f[1] = 1;

    for (i = 2; i <= n; i++) {
        f[i] = f[i - 1] + f[i - 2];
    }
    return f[n];
}
```

Is the algorithmic time complexity here same as the previous case $O(n^2)$? If yes, explain, if not then determine and explain what it will be.

Q5.[Total Marks:5+5+5=15]

Q5a.)

Describe the basic principle of working of quick sort. Your description must explain the idea of pivot, what computation is carried out with the pivot and a pseudo code that shows how an unsorted array of integers gets sorted. You need not write a complete C function.

Q5b.)

Give an example that demonstrates the worst case scenario of quick sort and using the example estimate and explain the worst case time complexity of quick sort. NO CODE NEEDED.

Q5c)

Given the integer array {30,20,70,10,50} trace the working of quick sort on this array showing clearly the arrangement of the numbers in the array after each major iteration. By major iteration we mean an iteration where the chosen pivot

has been put in its proper place. Use the left most element as the pivot. You need to show only the rearranged array after the major steps without any further explanation. Your answer should look like this:

Initial array: 30,20,70,10,50

Array after dealing with the first pivot: ----, ---, ---, ----,----

Etc....

Q6:[Total Marks: 6+2+5+2=15]

- a.) Write a C function that takes two sorted arrays of m and n integers and puts them sorted in a third array, You assume that the third array can hold at least $m+n$ integers. The arguments of the function are the three arrays, m and n .
- b.) Trace the working of the function in part a.) using two sorted arrays {30,60,70} and {50,90}
- c.) Write the recursive merge sort algorithm for an array of k elements using the above function.
- d.) Trace the merge sort algorithm in part c.) using the array {40,30,60,10,20,70,50}