Assignment 2

Deadline Friday 9th JULY 2021 (next week)

Answer all questions

a) Using the RAM model and the following code extract for sorting, derive the time complexity equation.

```
if( left + 10 <= right)
        {
        int i = left, j = right - 1;
        for ( ; ; )
          {
            while (a[++i] < pivot) {}
            while (pivot < a[--j] ) {}

                if (i < j)
                swap (a[i],a[j]);
             else  break;
          }
        swap (a[i], a[right-1);
        quicksort ( a, left, i-1);
        quicksort (a, i+1, right);
      }
else  insertionsort (a, left, right);
```

b) Using an example of a graph, briefly describe the operation of Depth First Traversal (DFS)

c) The master method solves recurrences of the form $T(n)=aT(n/b)+f(n)$ where $a>=1$ and $b>1$ are constants and $f(n)$ is a asymptotically positive function .

i) To use the master method, explain the THREE cases.

If the recurrence equation is $T(n)=2T(n/2)+n\log n$, use the Master method to show it's closed form .

d) The following is a function of a mergesort algorithm;

```
mergesort( int a[], int left, int right)

{
        if (right > left)
        {
        middle = left + (right - left)/2;
        mergesort(a, left, middle);
        mergesort(a, middle+1, right);
        merge(a, left, middle, right);
        }
}
```

Show that this algorithm has asymptotic order $\mathbf{O}(n\log n)$

e) Use repeated substitution to find the time complexity of the function **recurse**. Verify your result using induction.

```
/* Assume only non-negative even values of n are passed in */
  void recurse(int n) {
      int i, total = 0;
      if (n == 0) return 1;
      for (i = 0; i < 4; i++) {
          total += recurse(n-2);
          }
      return total;
  }
```

Study the following search algorithms given below: i.e. *Linear or Sequential Search*, *Binary and Interpolation* Task:

i) Evaluate each one of them using Big O notation;

ii) If you were to consider implanting one of them, discuss the choice you would make for a large data set.

A. Linear or Sequential Search :

```
#include <stdio.h>
int main()
{ int array[100], search, c, n;
printf("Enter the number of elements in array\n");
scanf("%d",&n); printf("Enter %d integer(s)\n", n);
for (c = 0; c < n; c++)
scanf("%d", &array[c]);
printf("Enter the number to search\n");
scanf("%d", &search);
for (c = 0; c < n; c++)
 { if (array[c] == search)     /* if required element found */
{
     printf("%d is present at location %d.\n", search, c+1); break;
 }  }
if (c == n) printf("%d is not present in array.\n", search);
return 0; }
```

B. Binary Search :

```
#include <stdio.h>
 int main()
{
int c, first, last, middle, n, search, array[100];
 printf("Enter number of elements\n");
scanf("%d",&n);
printf("Enter %d integers\n", n);
for ( c = 0 ; c < n ; c++ )
scanf("%d",&array[c]);
printf("Enter value to find\n");
scanf("%d",&search);
first = 0; last = n - 1; middle = (first+last)/2; while(
first <= last )
{ if ( array[middle] < search ) first = middle + 1; else if (
array[middle] == search )
{
printf("%d found at location %d.\n", search, middle+1); break; } else last =
middle - 1; middle = (first + last)/2;
} if ( first > last )
printf("Not found! %d is not present in the list.\n",    search); return 0; }
```

C. Interpolation Search :

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int a[25],n,mid,low,high,f=0,item,i;
```

```
printf("Enter the size of the array");
scanf("%d",&n);
printf("Enter the elements in sorted order");
for(i=0;i<n;i++) { scanf("%d",&a[i]); }
printf("Enter the item to be searched for");
scanf("%d",&item); low=0; high=n-1;
while(low<=high) {
mid=(low+(high-low)*((item-a[low])/(a[high]-a[low]))); if(a[mid]==item) {
printf("\n\nItem found at position %d",mid);
f=1; break; } else if(a[mid]>item)
{ high=mid-1; } else
{
low=mid+1; }} if(f==0)
printf("\n\nItem not found in the array"); getch(); }
```