Week 4 :

Topic : Recursion and Recursion removal using stack, Median and Interpolation search

1)

a) if(number==0)

b) return(number + mystery(number - 1));

c) Any nonnegative integer.

d)YES

e)YES

f) No as there is no base case for number <0

1. Write a recursive algorithm to multiply two positive integers m and n using repeated addition. Specify the base case and the recursive case.

int multiply(intn,int m)

{

If(n==0)

Return 0;

If(n==1)

Return m;

Return(m+multiply(m,n-1));

}

3) Given a sorted array with possibly duplicate elements,rite an efficient code to find the first and last position of the repeated element in the array. If element is found then code should return the index. If element not found then code should -1,-1;

For example:

Example 1:

Array A= [5, 7, 7, 8, 8, 10] Key= 8 Output= [3, 4]

Array A= [5, 7, 7, 8, 8, 10] Key= 6 Output= [-1, -1]

#include <bits/stdc++.h>

using namespace std;

int first\_occ(int arr[], int low, int high, int x, int n)

{

if (high >= low) {

int mid = low + (high - low) / 2;

if ((mid == 0 || x > arr[mid - 1]) && arr[mid] == x)

return mid;

else if (x > arr[mid])

return first\_occ(arr, (mid + 1), high, x, n);

else

return first\_occ(arr, low, (mid - 1), x, n);

}

return -1;

}

int last\_occ(int arr[], int low, int high, int x, int n)

{

if (high >= low) {

int mid = low + (high - low) / 2;

if ((mid == n - 1 || x < arr[mid + 1]) && arr[mid] == x)

return mid;

else if (x < arr[mid])

return last\_occ(arr, low, (mid - 1), x, n);

else

return last\_occ(arr, (mid + 1), high, x, n);

}

return -1;

} int main()

{

int arr[] = { 10, 20, 20, 30, 20, 30, 40, 70, 80, 80 };

int n = sizeof(arr) / sizeof(int);

int x = 20;

cout<<"First Occurrence of 20:"<<first\_occ(arr, 0, n - 1, x, n);

cout<<"\nLast Occurrence of 20:"<<last\_occ(arr, 0, n - 1, x, n);

return 0;

}

