

Algorithms and Problem-Solving Lab (15B17CI471)

EVEN 2022

Week -2 (14 Feb - 19 Feb 2022)

1. You are given an array $A[m]$ where first n cells contain integers in sorted order and the rest of the cells are filled with 0. Here assumes $m \gg n$ and value of n is unknown. Implement an algorithm that takes an integer x as input and finds a position in the array containing x , if such a position exists, in $O(\log n)$ time.

```
#include <iostream>
using namespace std;

int binarySearch(int *arr, int a, int b, int x){
    if(a>b){
        return -1;
    }
    int mid = a + (b-a)/2;
    if(arr[mid]==x){
        return mid;
    }
    if(arr[mid]==0 || arr[mid]>x){
        return binarySearch(arr, a, mid-1, x);
    } else {
        return binarySearch(arr, mid+1, b,x);
    }
    return -1;
}

int main(){
    // int arr[] = {1,2,3,4,5,7,8,9,0,0,0,0,0,0,0,0,0,0,0,0,0};
    // cout << binarySearch(arr, 0, sizeof(arr)/sizeof(int)-1, 6);
    int m,n,key;
    cin >> m >> n;
    int *arr = new int[m];
    for(int i = 0; i < n; i++){
        cin >> arr[i];
    }
    cin >> key;
    cout << binarySearch(arr, 0,m-1,key);

    return 0;
}
```

```

100
5
1 3 4 5 6
3
1
Process returned 0 (0x0)   execution time : 6.621 s
Press any key to continue.

```

2. Find the complexity of the following code snippets:

<p>(a) Function: One ()</p> <pre> { int x; int i; int n; x = 20; input n; for(i = 0; i < n; i++) x++; output x; } </pre>	<p>(b) Function: Two (int n)</p> <pre> { int *x; int i; allocate memory for x to store n elements for(i = 0; i < n; i++) { input x[i]; x[i] = x[i] + i; output x[i]; } } </pre>	<p>(c) Function: Three (int n)</p> <pre> { int *x; int i; allocate memory for x to store n elements for(i = 0; i < n; i++) { input x[i]; for(i = 0; i < n; i++) x[i] = x[i] + i; output x[i]; } } </pre>
<p>(d) Function: Four (int n, int y)</p> <pre> { int *x; int i; int j; j = 0; allocate memory for x to store n elements for(i = 0; i < n; i++) input x[i]; for(i = 0; i < n; i++) { if(x[i] == y) j++; } if(j > 0) output y is present j times else output y is not present } </pre>	<p>(e) Function: Five (int n)</p> <pre> { int *x; int i; int j; int m; int t; allocate memory for x to store n elements for(i = 0; i < n; i++) input x[i]; for(i = 0; i < n; i++) { m = x[i]; t = i; for(j = i+1; j < n; j++) { if(m > x[j]) { m = x[j]; t = j; } } x[t] = x[i]; x[i] = m; } for(i = 0; i < n; i++) output x[i]; } </pre>	<p>(f) Function: Six (int m, int n)</p> <pre> { int **x; int i; int j; int s; allocate memory for x to store m*n elements s = 0; for(i = 0; i < n; i++) { for(j = 0; j < n; j++) input x[i][j]; } for(i = 0; i < n; i++) { for(j = 0; j < n; j++) s = s + input x[i][j]; } output s; } //formulate the required algorithmic time for above function when (a) m = n (b) m ≠ n </pre>

- A. $O(n)$
- B. $O(n)$
- C. $O(n)$
- D. $O(n)$
- E. $O(n^2)$
- F. $O(m^2)$, $O(m*n)$

3. Implement the recursive algorithms for (a) Tower of Hanoi and (b) Fibonacci Number computation and analyse the space and time requirements of both the algorithms.

```
#include <bits/stdc++.h>
using namespace std;

void towerOfHanoi(char from, char to, char aux, int n){
    if(n==0){
        return;
    }
    towerOfHanoi(from, aux, to, n-1);
    cout << "Move " << n << " from " << from << " to " << to << "\n";
    towerOfHanoi(aux, to, from, n-1);
}

int fibonacci(int n){
    if(n==1){
        return 0;
    } else if(n==2){
        return 1;
    } else {
        return fibonacci(n-1) + fibonacci(n-2);
    }
}

int main()
{
    towerOfHanoi('a', 'c', 'b', 3);
    for (int i = 1; i < 10; i++)
    {
        cout << fibonacci(i) << " ";
    }

    return 0;
}
```

```

Move 1 from a to c
Move 2 from a to b
Move 1 from c to b
Move 3 from a to c
Move 1 from b to a
Move 2 from b to c
Move 1 from a to c
0 1 1 2 3 5 8 13 21

```

5. Implement the algorithm (Algo_1) presented below and discuss which task this algorithm performs. Also, analyse the time complexity and space complexity of the given algorithm. Further, implement the algorithm with following modification: replace $m = \lfloor 2n/3 \rfloor$ with $m = \lceil 2n/3 \rceil$, and compare the tasks performed by the given algorithm and modified algorithm.

```

Algo_1(A [0 ... n-1])
{
  if n = 2 and A[0] > A[1]
    swap A[0] ↔ A[1]
  else if n > 2
    m = ⌈ 2n/3 ⌉
    Algo_1 (A [0 .. m - 1])
    Algo_1 (A [n - m .. n - 1])
    Algo_1 (A [0 .. m - 1])
}

```

```

#include <bits/stdc++.h>
using namespace std;

void algo1(int *arr, int a, int b){
    int n = b-a+1;
    if(n==2 && arr[a]>arr[a+1]){
        int temp = arr[a];
        arr[a] = arr[a+1];
        arr[a+1] = temp;
    }
    if(n>2){
        int m = ceil((2*n)/3.0);
        algo1(arr, a, a+m-1);
        algo1(arr, a+n-m, b);
        algo1(arr, a, a+m-1);
    }
}

```

```

void algo2(int *arr, int a, int b){
    int n = b-a+1;
    if(n==2 && arr[a]>arr[a+1]){
        int temp = arr[a];
        arr[a] = arr[a+1];
        arr[a+1] = temp;
    }
    if(n>2){
        int m = floor((2*n)/3.0);
        algo2(arr, a, a+m-1);
        algo2(arr, a+n-m, b);
        algo2(arr, a, a+m-1);
    }
}

int main()
{
    int n;
    cin >> n;
    int *arr = new int[n];
    for(int i = 0; i < n; i++){
        cin >> arr[i];
    }
    algo2(arr, 0, n-1);
    for(int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
    return 0;
}

```

```

6
3 8 9 1 2 7
1 2 3 7 8 9

```

```

6
3 8 9 1 2 7
3 8 1 9 2 7

```

Algo 1 does sorting of array, Algo 2 does not do any significant task on the array.
Time Complexity of algorithm is

Page:-

$$T(n) = 1 + 3T\left(\frac{2n}{3}\right)$$

$$T(n) = 1 + 3\left[1 + 3T\left(\frac{2 \times 2n}{3 \times 3}\right)\right]$$

$$= 1 + 3 + 9T\left(\frac{4n}{9}\right)$$

$$= 1 + 3 + 9\left[1 + 3T\left(\frac{8n}{27}\right)\right]$$

$$T(n) = 1 + 3 + 9 + 27T\left(\frac{8n}{27}\right)$$

$$= (3^0 + 3^1 + 3^2 + \dots) + 3^i T\left(\left(\frac{2}{3}\right)^i n\right)$$

$$\left(\frac{2}{3}\right)^i n = 1$$

$$n = \left(\frac{3}{2}\right)^i$$

$$\log_{3/2} n = i$$

$$T(n) = 3^0 + 3^1 + 3^2 + 3^3 + \dots + 3^i$$

$$= \frac{1(3^{i+1} - 1)}{3 - 1}$$

$$= \frac{3^{\log_{3/2} n} - 1}{2}$$