Lec-31
02/06/24 11:14 AM

(i) Tower of Hanoi

n = 3

Hint

Source

src: A hp: B dest: C

1: Move n-1 disks from src to hp

2: Move the remaining disk from src to dest

3: Move n-1 disks from hp to dest

Code (3) (1) Step wise

src: A hp: B dest: C

Step 1 Step 2 Step 3

Step 4 Step 5 Step 6

Step 7

Step 8 Step 9

Move 1st disk from A to C
Move 2nd disk from A to B
Move 1st disk from C to B
Move 3rd disk from A to C
Move 1st disk from B to A
Move 2nd disk from B to C
Move 1st disk from A to C

```
Public static void TOH(int n, String src, String hlp, String des) {
    if(n == 0) {
        return;
    }
    TOH(n-1, src, des, hlp);
    System.out.println("Move " + n + "th disk from " + src + " to " + des);
    TOH(n-1, hlp, src, des);
}
```

n = 3

src: A hp: B des: C

Step 1: TOH(2, A, B, C)

Step 2: TOH(1, A, C, B)

Step 3: TOH(1, B, A, C)

Step 4: Move 3rd disk from A to C

Step 5: TOH(2, B, C, A)

Step 6: TOH(1, B, C, A)

Step 7: TOH(1, A, B, C)

Step 8: Move 2nd disk from A to C

Step 9: TOH(1, B, A, C)

Step 10: Move 1st disk from B to C

Step 11: Move 1st disk from A to C

Step 12: Move 1st disk from B to A

Step 13: Move 2nd disk from B to C

Step 14: Move 1st disk from A to C

divide

[2, 3, 5, 1, 8, 3, 4] Sorted

[1, 0, 6, 9, 10, 15, 17, 8, 19] Sorted

merge Sinde Array (Sorted)

```
public static int[] Merge(int[] arr1, int[] arr2) {
    int[m arr1.length;
    int[n arr2.length;
    int[] ans= new int[m+n];
    int i= 0;
    int j= 0;
    int k= 0;
    while(i< m && j< n) {
        if(arr1[i] < arr2[j]) {
            ans[k] = arr1[i];
            i++;
            k++;
        }
        else{
            ans[k] = arr2[j];
            j++;
            k++;
        }
    }
    while(i< m) {
        ans[k] = arr1[i];
        i++;
        k++;
    }
    while(j< n) {
        ans[k] = arr2[j];
        j++;
        k++;
    }
    return ans;
}
```

m = 4
n = 5

[2, 5, 14, 15]

[1, 3, 7, 19, 4]

[1, 2, 3, 3, 4, 5, 7, 8, 9]

Merge Sort

Divide

Merge

Sorted

Problem

Sorted

Sorted

```
Public static int Partition(int[] arr, int si, int ei) {
    int temp = arr[ei];
    int idx = si;
    for(int i = si; i< ei; i++) {
        if(arr[i] <= item) {
            int temp= arr[idx];
            arr[idx] = arr[i];
            arr[i] = temp;
            idx++;
        }
    }
    int temp = arr[idx];
    arr[idx] = arr[ei];
    arr[ei] = temp;
    return idx;
}
```

item = 4
idx = 2

[5, 2, 0, 3, 8, 1, 4]

[2, 1, 5, 3, 8, 1, 4]

[2, 3, 5, 3, 8, 1, 4]

[2, 3, 1, 5, 8, 3, 4]

[2, 3, 1, 4, 8, 5, 3]

Sorted Array

Sorting

Searching

Binary Search → Recursion

Merge Sort

Quick Sort

Stack

Queue

Linked List

Array

Stack

Queue

Linked List

Array

Stack

Queue

Linked List

Array