BML MUNJAL UNIVERSITY

DSA LAB ASSIGNMENT

Name: Manikanta Mandala.

Sec: CSE-3.

Enrolment No. : 200C2030134

Problem Statement:

Write a program to solve the problem of Tower of Hanoi.

Number of Towers and number of rings (use necessary conditional statement to satisfy the conditions)

Rules

The mission is to move all the disks to some another tower without violating the sequence of arrangement. A few rules to be followed for Tower of Hanoi are −

- Only one disk can be moved among the towers at any given time.
- Only the "top" disk can be removed.
- No large disk can sit over a small disk.

Solution:

If we use recursive method, basically we have three steps in Tower of Hanoi:

STEP1:

Make the largest disc alone in the source tower.

STEP2:

Move the largest disc from source tower to destination tower.

STEP3:

Fill again the destination tower with all other towers

Here the base condition is:

If the move the largest disc to the destination tower in that sub-array(kind off).

Code:

Tower Of Hanoi Class:

```java
package learningDSA;

public class towerOfHanoi {
    void process(int n,String src,String helper,String des){
        if(n==1){
            System.out.println("transfer disc "+n+" from "+src+" to "+des);

            return;
        }
        process(n-1,src,des,helper);
        System.out.println("transfer disc "+n+" from "+src+" to "+des);
        process(n-1,helper,src,des);

    }
}
```

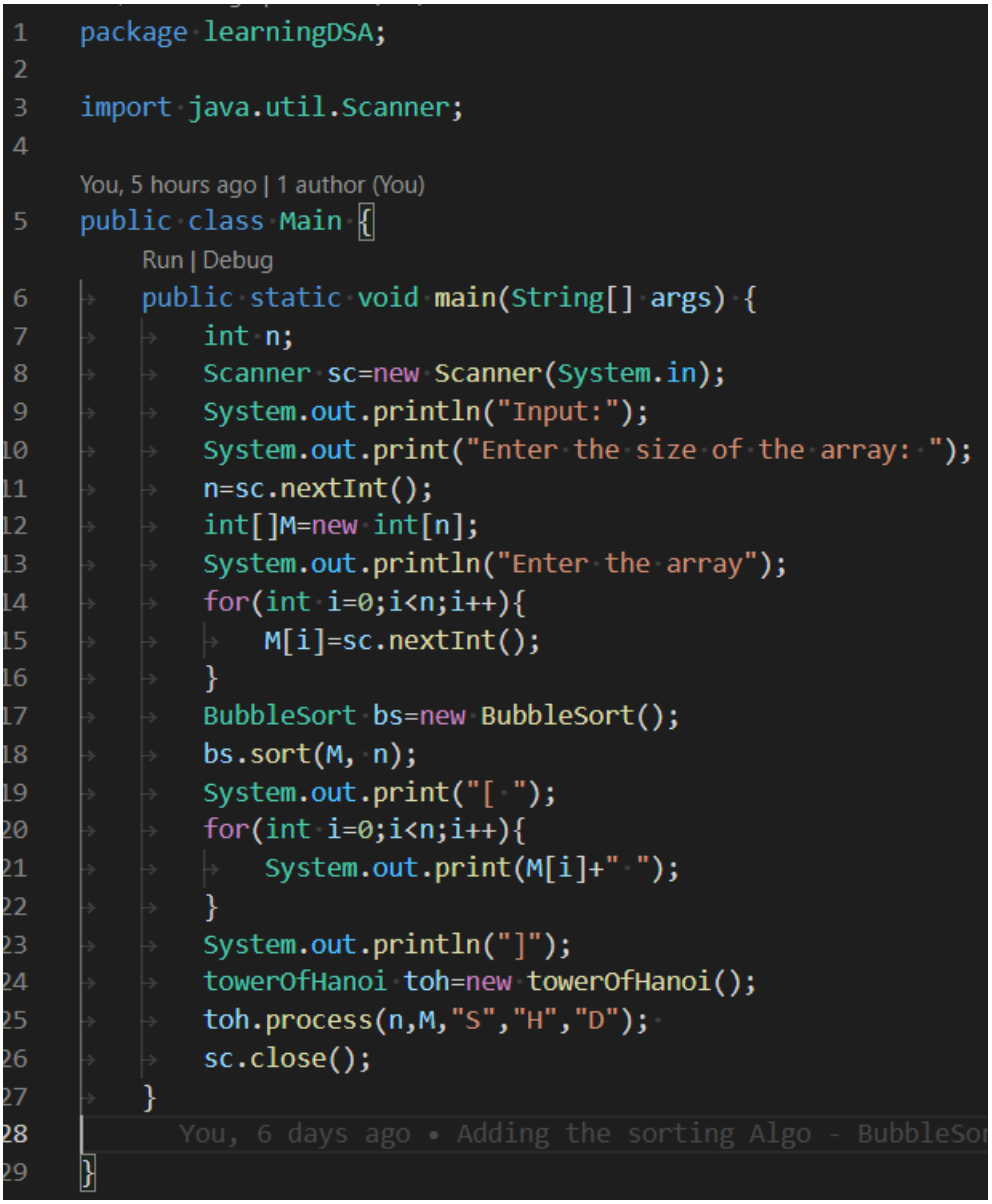
Bubble Sort Class:

```java
package learningDSA;
You, 5 hours ago | 1 author (You)
public class BubbleSort {
    static int i=1;      You, 5 hours ago • Adding sorting to Tower of Hanoi problem
    void sort(int[] M,int n) {

        if(n==1){
            System.err.println();
            return;
        }
        System.out.println("Iteration "+i);
        int swap = 0;
        for(int j=0;j<n-1;j++) {
            if(M[j+1]<M[j]) {
                int temp=M[j+1];
                M[j+1]=M[j];
                M[j]=temp;
                swap++;
                System.out.println("Items compared: ["+ M[j+1]+","+M[j]+" ] => swapped ["+ M[j]+","+M[j+1]+" ]" );
            }
            else{
                System.out.println("Items compared: ["+ M[j]+","+M[j+1]+" ] => not swapped" );
            }
        }
        if(swap==0){
            System.out.println();
            return;
        }
        i++;
        sort(M,n-1);

    }
}
```

Main class:



```java
package learningDSA;

import java.util.Scanner;

You, 5 hours ago | 1 author (You)
public class Main {
    Run | Debug
    public static void main(String[] args) {
        int n;
        Scanner sc=new Scanner(System.in);
        System.out.println("Input:");
        System.out.print("Enter the size of the array: ");
        n=sc.nextInt();
        int[]M=new int[n];
        System.out.println("Enter the array");
        for(int i=0;i<n;i++){
            M[i]=sc.nextInt();
        }
        BubbleSort bs=new BubbleSort();
        bs.sort(M, n);
        System.out.print("[ ");
        for(int i=0;i<n;i++){
            System.out.print(M[i]+" ");
        }
        System.out.println("]");
        towerOfHanoi toh=new towerOfHanoi();
        toh.process(n,M,"S","H","D");
        sc.close();
    }
    You, 6 days ago • Adding the sorting Algo - BubbleSor
}
```

Input and Output: