

14/11/2024

DSA PRACTICE 4

1. Stock Buy and Sell

```
import java.util.Scanner;
```

```
class Solution {
```

```
    public int maximumProfit(int prices[]) {  
        int minimum = prices[0];  
        int res = 0;  
        int n = prices.length;  
        for (int i = 1; i < n; i++) {  
            minimum = Math.min(minimum, prices[i]);  
            res = Math.max(res, prices[i] - minimum);  
        }  
        return res;  
    }  
}
```

```
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int[] prices = new int[n];  
        for (int i = 0; i < n; i++) {  
            prices[i] = sc.nextInt();  
        }  
        Solution sol = new Solution();
```

```

        System.out.println(sol.maximumProfit(prices));

        sc.close();
    }
}

```

Time Complexity : $O(n)$

```

C:\Users\P00JA\Documents\DSA_Practice4>javac BuyAndSellStock.java

C:\Users\P00JA\Documents\DSA_Practice4>java BuyAndSellStock
6
7 1 5 3 6 4
5

```

2.Coin Change(count ways)

```
import java.util.Scanner;
```

```

class GFG {

    static int count(int coins[], int n, int sum) {

        if (sum == 0)

            return 1;

        if (sum < 0)

            return 0;

        if (n <= 0)

            return 0;

        return count(coins, n - 1, sum) + count(coins, n, sum - coins[n - 1]);

    }

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();
    }
}

```

```

        int coins[] = new int[n];

        for (int i = 0; i < n; i++) {

            coins[i] = sc.nextInt();

        }

        int sum = sc.nextInt();

        System.out.println(count(coins, n, sum));

        sc.close();

    }
}

```

Time complexity : $O(\text{sum})$

```

C:\Users\P00JA\Documents\DSA_Practice4>javac CoinChange.java

C:\Users\P00JA\Documents\DSA_Practice4>java CoinChange
3
1 2 3
4
4

```

3.First and Last Occurrences

```

import java.util.ArrayList;

import java.util.Scanner;

```

```

class FirstAndLastOccurrence {

    ArrayList<Integer> find(int arr[], int x) {

        int n = arr.length;

        int first = -1, last = -1;

        for (int i = 0; i < n; i++) {

            if (arr[i] == x) {

                if (first == -1) {

                    first = i;

```

```
        }  
        last = i;  
    }  
}
```

```
    ArrayList<Integer> result = new ArrayList<>();  
    result.add(first);  
    result.add(last);  
    return result;  
}
```

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
  
    System.out.println("Enter the number of elements in the array:");  
    int n = sc.nextInt();  
  
    int[] arr = new int[n];  
  
    System.out.println("Enter the elements of the array:");  
    for (int i = 0; i < n; i++) {  
        arr[i] = sc.nextInt();  
    }  
  
    System.out.println("Enter the element to find:");  
    int x = sc.nextInt();  
  
    FirstAndLastOccurrence res = new FirstAndLastOccurrence();
```

```

        ArrayList<Integer> result = res.find(arr, x);

        if (result.get(0) == -1) {
            System.out.println("Element not found");
        } else {
            System.out.println("First Occurrence = " + result.get(0));
            System.out.println("Last Occurrence = " + result.get(1));
        }

        sc.close();
    }
}

```

Time Complexity : $O(n)$

```

C:\Users\P00JA\Documents\DSA_Practice4>javac FirstAndLastOccurrence.java

C:\Users\P00JA\Documents\DSA_Practice4>java FirstAndLastOccurrence
Enter the number of elements in the array:
9
Enter the elements of the array:
1 3 5 5 5 67 123 125
Enter the element to find:
5
First Occurrence = 2
Last Occurrence = 5

```

4.Find Transition Point

```

import java.util.Scanner;

class TransitionPoint
{
    static int findTransitionPoint(int arr[], int n)
    {

```

```
        for(int i = 0; i < n; i++)  
            if(arr[i] == 1)  
                return i;  
  
        return -1;  
    }  
}
```

```
public static void main (String[] args)  
{  
    Scanner sc = new Scanner(System.in);  
  
    System.out.println("Enter the number of elements in the array:");  
    int n = sc.nextInt();  
  
    int arr[] = new int[n];  
  
    System.out.println("Enter the elements of the array (0s and 1s only):");  
    for (int i = 0; i < n; i++) {  
        arr[i] = sc.nextInt();  
    }  
  
    int point = findTransitionPoint(arr, n);  
  
    if (point >= 0)  
        System.out.println("Transition point is " + point);  
    else  
        System.out.println("There is no transition point");  
}
```

```

        sc.close();
    }
}

```

Time Complexity : $O(n)$

```

C:\Users\POOJA\Documents\DSA_Practice4>javac TransitionPoint.java

C:\Users\POOJA\Documents\DSA_Practice4>java TransitionPoint
Enter the number of elements in the array:
6
Enter the elements of the array (0s and 1s only):
0 0 0 0 1 1
Transition point is 4

```

5.Find Repeating Element

```
import java.util.HashSet;
```

```
import java.util.Scanner;
```

```

class RepeatingElement {
    static void printFirstRepeating(int arr[]) {
        int min = -1;
        HashSet<Integer> set = new HashSet<>();

        for (int i = arr.length - 1; i >= 0; i--) {
            if (set.contains(arr[i]))
                min = i;
            else
                set.add(arr[i]);
        }

        if (min != -1)
            System.out.println("The first repeating element is " + arr[min]);
    }
}

```

```

        else

            System.out.println("There are no repeating elements");

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of elements in the array:");

        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.println("Enter the elements of the array:");

        for (int i = 0; i < n; i++) {

            arr[i] = sc.nextInt();

        }

        printFirstRepeating(arr);

        sc.close();

    }

}

```

Time Complexity : $O(n)$

```

C:\Users\P00JA\Documents\DSA_Practice4>javac RepeatingElement.java

C:\Users\P00JA\Documents\DSA_Practice4>java RepeatingElement
Enter the number of elements in the array:
7
Enter the elements of the array:
10 5 3 4 3 5 6
The first repeating element is 5

```


6.Remove Duplicates Sorted Array

```
import java.util.HashSet;
```

```
import java.util.Scanner;
```

```
class RemoveDuplicates{
```

```
    static int removeDuplicates(int[] arr) {  
        HashSet<Integer> s = new HashSet<>();  
        int idx = 0;  
  
        for (int i = 0; i < arr.length; i++) {  
            if (!s.contains(arr[i])) {  
                s.add(arr[i]);  
                arr[idx++] = arr[i];  
            }  
        }  
  
        return idx;  
    }  
}
```

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
  
    System.out.println("Enter the number of elements in the array:");  
    int n = sc.nextInt();  
  
    int[] arr = new int[n];
```

```

        System.out.println("Enter the elements of the array:");

        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        int newSize = removeDuplicates(arr);

        System.out.println("Array after removing duplicates:");

        for (int i = 0; i < newSize; i++) {
            System.out.print(arr[i] + " ");
        }

        sc.close();
    }
}

```

Time Complexity : $O(n)$

```

C:\Users\P00JA\Documents\DSA_Practice4>javac RemoveDuplicates.java

C:\Users\P00JA\Documents\DSA_Practice4>java RemoveDuplicates
Enter the number of elements in the array:
9
Enter the elements of the array:
1 2 2 3 4 4 4 5 5
Array after removing duplicates:
1 2 3 4 5

```

7.Maximum Index

```

import java.util.Scanner;

public class FindMaximum {

```

```

int maxIndexDiff(int arr[], int n) {
    int maxDiff = -1;
    for (int i = 0; i < n; ++i) {
        for (int j = n - 1; j > i; --j) {
            if (arr[j] > arr[i] && maxDiff < (j - i))
                maxDiff = j - i;
        }
    }
    return maxDiff;
}

```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.println("Enter the number of elements in the array:");
    int n = sc.nextInt();

    int arr[] = new int[n];
    System.out.println("Enter the elements of the array:");
    for (int i = 0; i < n; i++) {
        arr[i] = sc.nextInt();
    }

    FindMaximum max = new FindMaximum();
    int maxDiff = max.maxIndexDiff(arr, n);
    System.out.println("The maximum index difference is: " + maxDiff);
}

```

```
        sc.close();
    }
}
```

Time Complexity : $O(n^2)$

```
C:\Users\P00JA\Documents\DSA_Practice4>javac FindMaximum.java
C:\Users\P00JA\Documents\DSA_Practice4>java FindMaximum
Enter the number of elements in the array:
10
Enter the elements of the array:
9 2 3 4 5 6 7 8 18 0
The maximum index difference is: 8
```

8.Wave Array

```
import java.util.Arrays;
```

```
import java.util.Scanner;
```

```
class SortWave {
```

```
    void swap(int arr[], int a, int b) {
```

```
        int temp = arr[a];
```

```
        arr[a] = arr[b];
```

```
        arr[b] = temp;
```

```
    }
```

```
    void sortInWave(int arr[], int n) {
```

```
        Arrays.sort(arr);
```

```
        for (int i = 0; i < n - 1; i += 2)
```

```
            swap(arr, i, i + 1);
```

```
    }
```

```

public static void main(String args[]) {

    Scanner sc = new Scanner(System.in);


    System.out.println("Enter the number of elements in the array:");

    int n = sc.nextInt();


    int arr[] = new int[n];

    System.out.println("Enter the elements of the array:");

    for (int i = 0; i < n; i++) {

        arr[i] = sc.nextInt();

    }


    SortWave ob = new SortWave();

    ob.sortInWave(arr, n);


    System.out.println("Array in wave form:");

    for (int i : arr)

        System.out.print(i + " ");


    sc.close();

}
}

```

Time Complexity : $O(n \log n)$

```
C:\Users\P00JA\Documents\DSA_Practice4>javac SortWave.java
```

```
C:\Users\P00JA\Documents\DSA_Practice4>java SortWave
```

```
Enter the number of elements in the array:
```

```
7
```

```
Enter the elements of the array:
```

```
10 90 49 2 1 4 3
```

```
Array in wave form:
```

```
2 1 4 3 49 10 90
```