https://course.acciojob.com/idle?question=e3d6077b-afee-46c2-85fe-58848e357910

- EASY
- Max Score: 30 Points

Subarray Sum Zero

Write a program to print starting and ending indexes of all the subarrays of `arr[]` whose sum is equal to zero.

If no valid subarray exists whose sum is zero then print -1.

Print all the indexes in increasing order of the starting index and if starting index is the same then print in the increasing order of the ending index.

Input Format

First line contains an integer `N` denoting size of the array.

Second line contains `N` integers denoting the array elements

Output Format

Print the starting and ending indexes of all the subarrays whose sum is zero as shown in the sample test case.

If no valid subarray is present then print `-1`.

```
Subarray found from Index i to j
```

Example 1

Input

```
10
3 4 -7 3 1 3 1 -4 -2 -2
```

Output

```
Subarray found from Index 0 to 2
Subarray found from Index 0 to 9
Subarray found from Index 1 to 3
Subarray found from Index 2 to 5
Subarray found from Index 3 to 9
Subarray found from Index 5 to 7
```

Explanation

Adding the given subarray values we can see that we get sum as zero for each subarray.

Example 2

Input

```
3
0 0 0
```

Output

```
Subarray found from Index 0 to 0
Subarray found from Index 0 to 1
Subarray found from Index 0 to 2
Subarray found from Index 1 to 1
Subarray found from Index 1 to 2
Subarray found from Index 2 to 2
```

Explanation

Adding the given subarray values we can see that we get sum as zero for each subarray.

Constraints

```
1 <= N <= 10^3
```

```
-10^6 <= arr[i] <= 10^6
```

Topic Tags

- Loops
- Arrays

My code

```
// n java
import java.util.*;

class Solution {
    public void zeroSubarray(int[] arr) {
        //Write code here and print output here
            int n=arr.length;
                int psarr[] = new int[n];
                psarr[0] = arr[0];
                for(int i=1;i<n;i++)
                        psarr[i] = psarr[i-1] + arr[i];

                HashMap<Integer, ArrayList<Integer>> hm = new
HashMap<>();
                ArrayList<ArrayList<Integer>> l1 = new ArrayList<>();

                for(int i=0;i<n;i++){
                        if(hm.containsKey(psarr[i]))
                        {
                                for(int j =0; j< hm.get(psarr[i]).size(); j++)
                                {
                                        ArrayList<Integer> temp = new
ArrayList<>();
                                        temp.add(hm.get(psarr[i]).get(j)+1);
```

```
                        temp.add(i);
                        l1.add(temp);
                    }

                }
                    // System.out.println("Subarray found from
Index "+ (hm.get(psarr[i])+1) +" to "+ (i));

                 if(psarr[i] == 0)
                 {
                    ArrayList<Integer> temp = new ArrayList<>();
                    temp.add(0);
                    temp.add(i);
                    l1.add(temp);
                 }
                    // System.out.println("Subarray found from
Index 0 to "+ i);
                if(hm.containsKey(psarr[i]))
                {
                    ArrayList temp2 = hm.get(psarr[i]);
                    temp2.add(i);
                    hm.put(psarr[i], temp2);
                }
                else
                {
                    ArrayList temp2 = new ArrayList<>();
                    temp2.add(i);
                    hm.put(psarr[i], temp2);
                }
```

```
            }
// Collections.sort(list, new Comparator<List<String>> () {
//     @Override
//     public int compare(List<String> a, List<String> b) {
//         return a.get(1).compareTo(b.get(1));
//     }
// });
            Collections.sort(l1, new Comparator<List<Integer>> () {
    @Override
    public int compare(List<Integer> a, List<Integer> b) {
            if(a.get(0)!= b.get(0))
            {
                if(a.get(0)<b.get(0))
                    return -1;
                else
                    return 1;
            }
            if(a.get(1)<b.get(1))
                    return -1;
                else
                    return 1;
            // return a.get(1)<b.get(1);

    }
});
            for(int i =0; i< l1.size(); i++)
            {
                System.out.println("Subarray found from Index
"+l1.get(i).get(0)+" to "+ l1.get(i).get(1));
            }
```

```
            if(l1.size()==0)
                System.out.println("-1");
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n;
        n = sc.nextInt();
        int arr[] = new int[n];
        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt();
        Solution Obj = new Solution();
        Obj.zeroSubarray(arr);
        sc.close();
            System.out.println();
    }
}
```

```
/*import java.util.*;
import java.lang.*;
import java.io.*;
```

```
public class Main
{
	public static void main (String[] args) throws
java.lang.Exception
	{
		//your code here
		Scanner s=new Scanner(System.in);
		int n=s.nextInt();
		long arr[]=new long[n];
		for(int i=0;i<n;i++)
			arr[i]=s.nextLong();
		for(int i=0;i<n;i++)
			{
				long sum=0;
				for(int j=i;j<n;j++)
					{
						sum+=arr[j];
						if(sum==0)
						{
							System.out.println("Subarray
found from Index "+i+" to "+j);
						}
					}
			}
	}
}*/
```