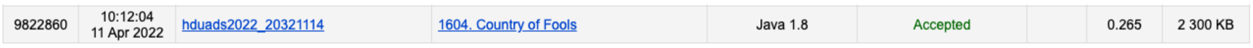
Laboratory work # 3

Student: HU Riqian
Student ID: 20321114
Timus Name: hduads2022_20321114

Mail: jhlxhrq@163.com

Problem # 1604. Country of Fools

Screenshot from Timus:



9822860	10:12:04 11 Apr 2022	hduads2022_20321114	1604. Country of Fools	Java 1.8	Accepted		0.265	2 300 KB

Explanation of algorithm:

1. Loop through the array and find the maximum, then let it stand with the second largest number.

2. Each time I choose the number, this number will be subtracted by 1 until no number in the array can be subtracted.

Computational complexity of algorithm:

$$O(n \log k)$$

Source code:

```
import java.io.*;

public class CountryOfFools {
    public static void main(String[] args) throws IOException {
        new CountryOfFools().run();
    }

    StreamTokenizer in;
    PrintWriter out;
    static int SIZE;
    static int[] SIGNS_NUMS;
    static int TOTAL_SIGNS = 0;
    static int prevSignIndex = -1;

    int nextInt() throws IOException {
        in.nextToken();
        return (int) in.nval;
    }

    void run() throws IOException {
        in = new StreamTokenizer(new BufferedReader(new
InputStreamReader(System.in)));
        out = new PrintWriter(System.out);
        inputSignPara();

        while (TOTAL_SIGNS > 0) {
            int maxSignIndex = findMaxIndex(SIGNS_NUMS, prevSignIndex);
            if (maxSignIndex == -1) {
                maxSignIndex = prevSignIndex;
            }
            SIGNS_NUMS[maxSignIndex] -= 1;
            prevSignIndex = maxSignIndex;
            System.out.print(maxSignIndex + 1 + " ");
            TOTAL_SIGNS--;
        }
        out.flush();
    }

    void inputSignPara() throws IOException {
        SIZE = nextInt();
        SIGNS_NUMS = new int[SIZE];
        for (int i = 0; i < SIZE; i++) {
            SIGNS_NUMS[i] = nextInt();
            TOTAL_SIGNS += SIGNS_NUMS[i];
        }
    }

    int findMaxIndex(int[] signs, int index) {
        int maxSign = 0;
        int result = -1;
        for (int i = 0; i < SIZE; i++) {
            if (i != index && signs[i] > maxSign) {
                maxSign = signs[i];
                result = i;
            }
        }
        return result;
    }
}
```