Algorithm Documentation

svtter

2018 06 24

Contents:

CHAPTER 1

POJ

1.1 POJ 1852

1.1.1 INPUT

2 10 3 2 6 7 214 7 11 12 7 13 176 23 191

1.1.2 OUTPUT

4 8 38 207

1.1.3 CODE

```
#include <iostream>
#include <stdio.h>
#include <string.h>
#include <cmath>

using namespace std;
```

(continues on next page)

()

```
int ants[1000001];
const int INF = 1000001;

void test()
{
    for (int i = 0; i < 3; i++)
    {
        cout << ants[i] << endl;
    }
}

void solve_problem()
{
    int pole, n;
    cin >> pole >> n;
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &ants[i]);
    }

    // scanf
    //test();

    int max = -INF;
    int dis;
    for (int i = 0; i < n; i++)
    {
        dis = pole - ants[i];
        dis = dis > ants[i] ? dis : ants[i];
        max = dis > max ? dis : max;
        //cout << "max: " << max << endl;
    }

    int min = 0;
    for (int i = 0; i < n; i++)
    {
        dis = pole - ants[i];
        dis = dis < ants[i] ? dis : ants[i];
        min = dis > min ? dis : min;
    }

    printf("%d %d\n", min, max);
}

int main()
{
    int cases;
    cin >> cases;

    while(cases--)
    {
        memset(ants, 0, sizeof(ants));
        solve_problem();
    }

    return 0;
```

(continues on next page)

()

```
}
```

CHAPTER 2

stlbinary_search

CHAPTER 3

Leetcode

3.1 Array

3.1.1 Array 1

from: https://leetcode-cn.com/explore/interview/card/top-interview-questions-easy/1/array/21/

C++ Source Code

```
class Solution {
public:
    int removeDuplicates(vector<int>& nums) {

        int length = nums.size();
        if (length == 0) {
            return 0;
        }
        int pre = nums[0];
        for (int i = 1 ;i < length; i ++) {
            if (pre == nums[i]) {
                nums.erase(nums.begin() + i);
                # cout << i << ' ' << nums[i];
                i --;
                length --;
            }
            else {
                pre = nums[i];
            }
        }
        return nums.size();
    }
};
```

Python Source Code

```
class Solution:
    def removeDuplicates(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        if len(nums) == 0:
            return 0
        pre = nums[0]

        for i in nums[1:]:
            if i == pre:
                nums.remove(i)
            else:
                pre = i
        return len(nums)
```

The problems about array.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search