


首页 > 笔经面经 > 网易2018校招内推笔试编程题参考代码及思路

网易2018校招内推笔试编程题参考代码及思路 精



NotDeep

编辑于 今天 18:26:34

回复 35 | 赞 17 | 浏览 5700

练习链接: <https://www.nowcoder.com/test/6291726/summary>

等差数列

分析

对序列排序,然后比对一下是否等差即可。

参考code

```
#include <bits/stdc++.h>

using namespace std;

int n;
int x[55];
string solve() {
    sort(x, x + n);
    if(n <= 2) return "Possible";
    else {
        int d = x[1] - x[0];
        bool ok = 1;
        for(int i = 1; ok && i < n; i++) {
            if(d != x[i] - x[i - 1]) return "Impossible";
        }
        return "Possible";
    }
}

int main() {
    cin >> n;
    for(int i = 0; i < n; i++) cin >> x[i];
    cout << solve() << endl;
    return 0;
}
```

彩色砖块

分析

题目有点唬人。。

设颜色种数为s,那么

if s > 2 ans = 0

else ans = s

参考代码

本文相关内容

Java工程师

C++工程师

iOS工程师

安卓工程师

运维工程师

前端工程师

算法工程师

热门推荐

帮你超越80%竞争对手的项目经验

帮你超越70%竞争对手的项目经验

手把手带你实现项目经验零突破

笔经面经近期热帖

【面经】阿里+百度+CVTE面经合集 ( offer均已收到 )

发表于 2016-05-23 03:16:39 回复 (373)

【这可能不只是一篇面经】

发表于 2017-07-22 17:49:01 回复 (76)

【数据挖掘面经】腾讯+百度+华为 ( 均拿到sp offer )

发表于 2016-10-13 23:10:29 回复 (67)

2017.8.1拼多多在线笔试题

发表于 2017-08-01 22:34:05 回复

令人绝望的华为南研所一面

发表于 2017-07-19 18:14:11 回复

评价下网易编程题难度

发表于 今天 17:03:31

那几个月在找工作 ( 百度, 华为 )

发表于 2015-10-15 16:41:06 回复 (33)

网易2018校招内推笔试编程题参考代码及思路

发表于 今天 17:28:58 回复 (35)



扫一扫下载牛客APP

刷真题、补算法、看面经、得内推

使用第三方账号直接登录使用吧: 更多

<https://www.nowcoder.com/discuss/32063?type=0&order=0&pos=9&page=2> 1/9

[登录](#) / [注册](#)

```
using namespace std;

string s;
set<char> S;
int main() {
    cin >> s;
    for(int i = 0; i < s.size(); i++) S.insert(s[i]);
    int ans = S.size();
    if(ans > 2) ans = 0;
    cout << ans << endl;
    return 0;
}
```

## 交错01串

### 分析

就挨着比较,记录当前是交错01串的长度,维护最大值即可。

### 参考代码

```
#include <bits/stdc++.h>

using namespace std;

string s;
int main() {
    cin >> s;
    int ans = 1, cnt = 1;
    for(int i = 1; i < s.size(); i++) {
        if(s[i] != s[i - 1]) {
            cnt++;
        } else {
            cnt = 1;
        }
        ans = max(ans, cnt);
    }
    cout << ans << endl;
    return 0;
}
```

## 独立的小易

### 分析

首先计算能保证的房屋能租多少天,如果有剩余再考虑购买水果。

### 参考代码

```
#include <bits/stdc++.h>

using namespace std;

int x, f, d, p;
int solve(int x, int f, int d, int p) {
```

发表于 2017-02-27 12:01:01 回复 (34)

携程状态更新了, , ,

发表于 今天 16:14:15 回复 (27)

```
    return f + d / (x + p);
}

int main() {
    cin >> x >> f >> d >> p;
    cout << solve(x, f, d, p) << endl;
    return 0;
}
```

## 堆棋子

### 分析

对于一个k,我们找一个坐标(x, y)让k个棋子距离这个坐标的曼哈顿距离之和最小。注意到x和y其实是独立的,考虑枚举棋盘上所有可能得坐标(x[i], y[j]),计算这个坐标到所有棋子的距离分别是多少,然后维护k个棋子对于这个坐标最小曼哈顿距离和即可。

### 参考代码

```
#include <bits/stdc++.h>

using namespace std;

const int inf = 1e9;

int n, x[55], y[55];
int main() {
    int n;
    cin >> n;
    for(int i = 0; i < n; i++) cin >> x[i];
    for(int i = 0; i < n; i++) cin >> y[i];
    vector<int> res(n, inf);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            for(int k = 0; k < n; k++) {
                vector<int> res2(n);
                for(int l = 0; l < n; l++) {
                    res2[l] = abs(x[l] - x[j]) + abs(y[l] - y[k]);
                }
                sort(res2.begin(), res2.end());
                int res3 = 0;
                for(int l = 0; l < i + 1; l++) res3 += res2[l];
                res[i] = min(res[i], res3);
            }
        }
    }
    for(int i = 0; i < n; i++) {
        i == 0 ? cout << res[i] : cout << " " << res[i];
    }
    return 0;
}
```

## 操作序列

### 分析

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;

int a[maxn];
int n;
int main() {
    scanf("%d", &n);
    for(int i = 1; i <= n; i++) scanf("%d", &a[i]);
    if(n % 2) {
        for(int i = n; i >= 1; i -= 2) printf("%d ", a[i]);
        for(int i = 2; i <= n; i += 2) i == n - 1 ? printf("%d", a[i]) : pr
printf("%d ", a[i]);
    } else {
        for(int i = n; i >= 1; i -= 2) printf("%d ", a[i]);
        for(int i = 1; i <= n; i += 2) i == n - 1 ? printf("%d", a[i]) : pr
printf("%d ", a[i]);
    }
    printf("\n");
    return 0;
}
```

## 疯狂队列

### 分析

根据样例提示,猜想了一个结论:

我们要把这个队列安排为交错的形式(证明略)。

交错有两种形式,看第一个人是比他相邻的人高还是矮。

以矮的为例:

总共的疯狂值为 $(h_2 - h_1) + (h_2 - h_3) + (h_4 - h_3) + \dots = -h_1 + 2h_2 - 2h_3 + 2h_4 - \dots$ ,

如果总共是偶数个人,疯狂值为 $-h_1 + 2h_2 - 2h_3 + 2h_4 - \dots - 2h_{n-1} + h_n$ ,所以我们需要从最大的开始依次安排 $h_2, \dots, h_{n-2}$ ,然后安排 $h_n$ ,然后继续安排剩下的。

如果总共是奇数个人,疯狂值为 $-h_1 + 2h_2 - 2h_3 + 2h_4 - \dots + 2h_{n-1} - h_n$ ,所以我们需要从最大的开始依次安排 $h_2, \dots, h_{n-1}$ ,然后安排 $h_1$ 和 $h_n$ ,然后继续安排剩下的。

然后另外一种形式类似,维护最大的疯狂值即可。

### 参考代码

```
#include <bits/stdc++.h>

using namespace std;

int h[55];
int n;
int main() {
    cin >> n;
    for(int i = 0; i < n; i++) cin >> h[i];
    sort(h, h + n);
    int tmp, ans = 0, ans1 = 0, ans2 = 0;
```

```
        ans += 2 * (h[tmp + i] - h[i]);
    }
    ans += h[tmp - 1] - h[tmp];
    cout << ans << endl;
    return 0;
} else {
    tmp = n / 2;
    for(int i = 0; i < tmp; i++) {
        ans1 += 2 * (h[tmp + 1 + i] - h[i]);
        ans2 += 2 * (h[tmp + 1 + i] - h[i]);
    }
    ans1 += -h[tmp] + h[tmp - 1];
    ans2 += h[n] - h[n + 1];
    cout << max(ans1, ans2) << endl;
    return 0;
}
return 0;
}
```

## 小易喜欢的数列

### 分析

想象一下,如果我们确定这个数列的第一个数是*i*,那么第二个数可以是1到*k*中除了是*i*的约数的任何数。  
于是我们定义dp[j][i]表示长度为*i*最后一个数是*j*的小易喜欢的数列的数量,然后挨着转移即可。。实现请参考参考代码。

### 参考代码

```
#include <bits/stdc++.h>

using namespace std;

const int mod = 1e9 + 7;
const int maxn = 1e5 + 5;
int dp[maxn][15];
int n, k;
int main() {
    cin >> n >> k;
    dp[1][0] = 1;
    for(int i = 1; i <= n; i++) {
        int sum = 0;
        for(int j = 1; j <= k; j++) {
            sum += dp[j][i - 1];
            sum %= mod;
        }
        for(int j = 1; j <= k; j++) {
            int sum2 = 0;
            for(int z = j + j; z <= k; z += j) {
                sum2 += dp[z][i - 1];
                sum2 %= mod;
            }
            dp[j][i] = (sum - sum2 + mod) % mod;
        }
    }
}
```

```
    ans += dp[j][n];
    ans %= mod;
}
cout << ans << endl;
return 0;
}
```

[收藏](#) | [分享](#) | [赞\(17\)](#) | [回帖](#)

## 35条回帖

[回帖](#)

米油牌小火切

21#

膜拜大佬！！！！向大佬学习！！！！

发表于 今天 17:48:19

[赞\(0\)](#)[回复\(0\)](#)

牛客7804650号

22#

这是哪个岗位的，前端考这些算法不

发表于 今天 17:50:06

[赞\(0\)](#)[回复\(0\)](#)

逍遥墨髓

23#

色块 题目，先撸了全排列，然后删去不同的，然后再对比。。简直sjb了。。

发表于 今天 17:50:34

[赞\(0\)](#)[回复\(1\)](#)[iamAKA](#) : 666是因为选择题刚做完全排列吗hhh吊

今天 17:57:14

[赞\(0\)](#)[回复](#)[回复](#)

柏青

24#

哎哎哎，还是要多做题啊

发表于 今天 17:52:53

[赞\(0\)](#)[回复\(0\)](#)

-\_-川呆子

25#

堆棋子这题想想复杂度就不敢写

发表于 今天 17:58:21

[赞\(1\)](#)[回复\(1\)](#)[生命不息、奋斗不止](#) : 淡淡的忧伤

今天 18:03:12

[赞\(0\)](#)[回复](#)[回复](#)

穷，你这么能做出来吗

发表于 今天 18:00:09

赞(0) 回复(0)



憨人Nonponder

27#

请问，棋盘格那个题目，为什么中心点的坐标一定是在已有的坐标里的呢？

发表于 今天 18:03:42

赞(0) 回复(1)

憨人Nonponder：想清楚了。。曼哈顿距离和欧式距离的差别。。还是要多多学习。。

今天 18:27:06

赞(0) 回复

请输入你的观点

回复



小新没有蜡笔~

28#

操作序列那一题，我和你的思路是一样的，只不过我用java实现，代码也基本差不多，可是只通过了

发表于 今天 18:03:44

赞(0) 回复(1)

小新没有蜡笔~：可是只通过了50%，说我超时。。。一直在疑通过50%后，说我超时，一直在疑惑

今天 18:05:02

赞(0) 回复

请输入你的观点

回复



太阳系帝王

29#

Java岗的各位都做出来了么？

发表于 今天 18:04:48

赞(0) 回复(0)



小淘气2

30#

大大大大。。。。大佬

发表于 今天 18:05:38

赞(0) 回复(0)



kohama

31#

操作队列，JAVA，也是只用一个数组，怎么超内存了。。。

发表于 今天 18:05:49

赞(0) 回复(0)



初識ゝ故人已

32#

棋盘那题你这种暴力解法在java会内存溢出的

发表于 今天 18:09:58

赞(0) 回复(0)



牛客9000057号

33#

堆棋子那个题 怎么保证 中心点就在你枚举的这些点里面呢

发表于 今天 18:10:51

赞(0) 回复(0)



bandoll

34#

难道题目上没写 2000

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

public class Main {
    public static Integer[] chess_min(Integer[] X, Integer[] Y, Integer n) {
        Set<Integer> xAxis = new HashSet<Integer>(Arrays.asList(X));
        Set<Integer> yAxis = new HashSet<Integer>(Arrays.asList(Y));

        List<List<Integer>> centers = new ArrayList<List<Integer>>();

        for (Integer x: xAxis) {
            for (Integer y: yAxis) {
                List<Integer> distances = new ArrayList<Integer>();
                for (Integer i=0; i<n; i++) {
                    distances.add(Math.abs(X[i]-x) + Math.abs(Y[i]-y));
                }
                Collections.sort(distances);
                centers.add(distances);
            }
        }

        Integer[] min_step = new Integer[n];
        Integer[] sum = new Integer[centers.size()];
        for (Integer i=0; i<centers.size(); i++) {
            sum[i] = 0;
        }
        for (Integer i=0; i<n; i++) {
            for (Integer j=0; j<centers.size(); j++) {
                sum[j] += centers.get(j).get(i);
            }
            min_step[i] = Collections.min(Arrays.asList(sum));
        }
        return min_step;
    }

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

        while(in.hasNext()) {
            Integer n = in.nextInt();
```



```
for (Integer i=0; i<n; i++) {
    X[i] = in.nextInt();
}
for (Integer i=0; i<n; i++) {
    Y[i] = in.nextInt();
}

StringBuilder str = new StringBuilder();
for (Integer i: chess_min(X, Y, n)) {
    str.append(i+" ");
}
System.out.println(str.substring(0, str.length()-1));
}
}
```

棋子 java过的

发表于 今天 18:21:22

[赞\(1\)](#)[回复\(0\)](#)[首页](#)[上一页](#)[1](#)[2](#)[下一页](#)[末页](#)

代码语言

[使用MD编辑器](#)

在这里畅所欲言你的看法吧！

[回帖](#)

牛客网，程序员必备求职神器

QQ群：169195721

微信：www\_nowcoder\_com

微博：牛客网

[关注](#)[关注](#)

扫一扫，把题目装进口袋

[关于我们](#) [加入我们](#) [意见反馈](#) [企业服务](#)[网站合作](#) [免责声明](#) [友情链接](#)

浙ICP备14000860号-2

京公网安备 11011402010078号

牛客网©2017 All Rights Reserved