

# 四月月赛题目讨论

## ➤ 问题1：数字统计

---

统计区间[L ,R]内2的个数

$$1 \leq L \leq R \leq 10000$$

## ➤ 问题1：数字统计

---

考察点：暴力，整数拆分

暴力枚举区间内每一个数，然后进行整数拆分数位，统计2出现的次数即可

## ➤ 问题1：数字统计

参考代码：

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

using namespace std;

int main() {
    freopen("two.in", "r", stdin);
    freopen("two.out", "w", stdout);
    int l, r, ans = 0;
    scanf("%d%d", &l, &r);
    for(int i = l; i <= r; i++)
        for(int k = i; k > 0; k /= 10)
            if(k % 10 == 2) ans++;
    printf("%d\n", ans);
    return 0;
}
```

## ➤ 问题2：接水问题

---

给定 $n$ 个人的接水所需时间 $w$ 以及水龙头 $m$ 的数量，求所有人接完水的时间。

$1 \leq n \leq 10000$ ,  $1 \leq m \leq 100$  且  $m \leq n$ ;  $1 \leq w_i \leq 100$ 。

## ➤ 问题2：接水问题

考察点: 模拟、贪心

假设每一个水龙头的出水量为 $L$ (初始化都为0),那么每一次输入一个 $w$ ,就去找出水量最小的水龙头。然后这个水龙头的出水量设置为 $L[i]+w$ 。最后找到 $L$ 的最大值,这个值就是答案。

为什么?这里假设每个人的接水时间全部算在水龙头上。每一次有人要接水时,必定是找目前出水量最小(即目前最快的一个水龙头),而最后那个人离开后程序才能算结束。所以要求最后那个人离开的时间,即出水量最大的水龙头。

## ➤ 问题2：接水问题

参考代码：

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

using namespace std;

int water[10005];
int main() {
    freopen("water.in", "r", stdin);
    freopen("water.out", "w", stdout);
    int min = 1, ans = -1;
    int n, m, w;
    scanf("%d%d", &n, &m);
    for(int i = 1; i <= n; i++) {
        scanf("%d", &w);
        min = 1;
        for(int j = 2; j <= m; j++) {
            if(water[j] < water[min]) {
                min = j;
            }
        }
        water[min] += w;
        ans = max(ans, water[min]);
    }
    printf("%d\n", ans);
    return 0;
}
```

## ➤ 问题3：导弹拦截

给定两个大点的坐标以及n个小点的坐标，使两个大点作为圆心所构成的圆的面积可以覆盖所有点，再这个条件下使半径的平方和最小。

对于100%的数据， $1 \leq N \leq 100000$ ，且所有坐标分量的绝对值都不超过1000。



## ➤ 问题3：导弹拦截

---

考察点：贪心

我们先按每一颗导弹到点A的距离排一个序，然后枚举某个点P，将这个点看做断点。即点P前的点归入点A中，点P后的点归入点B中。我们枚举一下点，然后求出最小的平方和即可。

## ➤ 问题3：导弹拦截

参考代码：

```
#include <bits/stdc++.h>
using namespace std;
const int maxn = 100002, inf = 1000000000;
int X1, Y1, X2, Y2, n;
struct data {
    int x, y;
    long long t1, t2;
};
bool cmp(data a, data b) {
    return a.t1 < b.t1;
}
data d[maxn];
int main() {
    freopen("missile.in", "r", stdin);
    freopen("missile.out", "w", stdout);
    scanf("%d%d%d%d", &X1, &Y1, &X2, &Y2);
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        scanf("%d%d", &d[i].x, &d[i].y);
        d[i].t1 = (long long)(X1 - d[i].x)*(X1 - d[i].x) + (long long)(Y1 - d[i].y)*(Y1 - d[i].y);
        d[i].t2 = (long long)(X2 - d[i].x)*(X2 - d[i].x) + (long long)(Y2 - d[i].y)*(Y2 - d[i].y);
    }
    sort(d + 1, d + n + 1, cmp);
    long long r2 = 0, ans = inf;
    for (int i = n; i > 0; i--) {
        r2 = max(d[i + 1].t2, r2);
        ans = min(ans, r2 + d[i].t1);
    }
    printf("%lld", ans);
    return 0;
}
```

## ➤ 问题4：三国游戏

---

给出 $n$ 个人的关系值，求你在不能完成最大搭配的情况下所能做到的最大搭配

更简练的意思：

给出 $n$ 个人的关系值，求每个人的次大关系值

## ➤ 问题4：三国游戏

---

考察点：贪心、博弈论

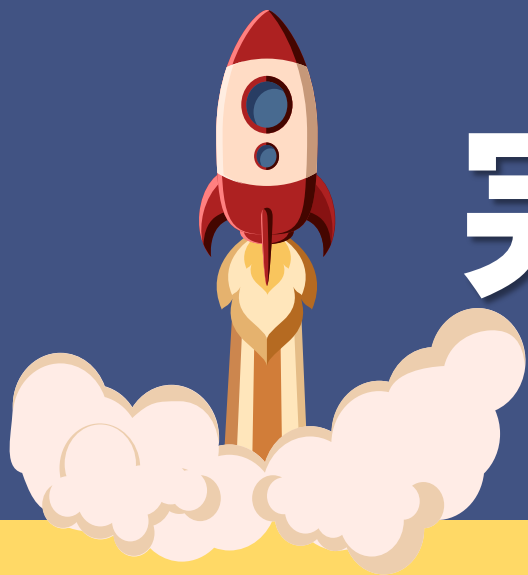
小涵是必胜的，取对于每个武将第二大的匹配值的最大值就是答案。

## ➤ 问题4：三国游戏

参考代码：

```
#include <bits/stdc++.h>
using namespace std;

int n, a[5005][5005], maxx, maa, ans;
int main() {
    freopen("sanguo.in", "r", stdin);
    freopen("sanguo.out", "w", stdout);
    scanf("%d", &n);
    for(int i = 1; i <= n; i++) {
        maxx = maa = 0; //注意清零
        for(int j = i + 1; j <= n; j++)
            scanf("%d", &a[i][j]), a[j][i] = a[i][j];
        for(int j = 1; j <= n; j++) {
            if (a[i][j] > maxx) {
                maa = maxx;
                maxx = a[i][j];
            } else
                maa = max(a[i][j], maa);
            ans = max(ans, maa);
        }
    }
    printf("1\n%d", ans);
}
```



# 完！

以梦为码 心之所往

