

进阶冲刺 期中奖金赛

<https://vjudge.net/contest/367558>

A - 为美好的世界献上智障女神

考察点

基本语法、细节

题目大意

n 只青蛙，每只生命为 k ，每分钟造成1点伤害，杀死一只青蛙后再攻击下一只，问 m 分钟后几只未被伤害过。

Solution

- 如果 m 恰好能被 k 整除，答案为 $n-m/k$ 。
- 如果 m 不能被 k 整除，答案为 $n-m/k-1$ ，因为还剩下几分钟造成了伤害但没有击杀。
- 如果上面的答案小于0，则输出0，而非负数。因为时间足够长的话所有青蛙都被击杀。

```
// a_mandrake 's solution for Problem A
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n,k,m,ans;
    cin>>n>>k>>m;
    if(m%k!=0) ans=n-m/k-1;
    else ans=n-m/k;
    if(n-m/k<=0) ans=0;
    cout<<ans;
    return 0;
}
```

B - 为美好的世界献上废柴店主

考察点

前缀和思想应用

题目大意

给出 n 天账目信息 $a[1..n]$ ， q 次询问，每次询问 p ，求 $a[1..p]$ 中最大值减最小值

Solution

前缀和思维应用，设 Max 、 Min 数组， $Max[i]$ 、 $Min[i]$ 分别记录前 i 天的最大值与最小值，对于询问输出即可。

```
//yggdyy_'s solution for Problem B
#include <iostream>
```

```

#include <cmath>
#include <cstdio>
#include <algorithm>

using namespace std;
int a[101010], maxx[101010], minn[101010];
int n, m;
int main(){
    maxx[0] = -999999999;
    minn[0] = 999999999;
    cin >> n >> m;
    for(int i=1; i<=n; i++) scanf("%d", &a[i]);
    for(int i=1; i<=n; i++){
        if(a[i] > maxx[i-1]) maxx[i] = a[i];
        else maxx[i] = maxx[i-1];

        if(a[i] < minn[i-1]) minn[i] = a[i];
        else minn[i] = minn[i-1];
    }
    for(int i=1; i<=m; i++){
        int qwq;
        scanf("%d", &qwq);
        printf("%d ", maxx[qwq] - minn[qwq]);
    }
    return 0;
}

```

C - 为美好的世界献上地下城主

考察点

搜索 (dfs/bfs), 剪枝

题目大意

10 * 10 的迷宫，左上角入口，左下角出口，中间必须经过某些点（宝箱）问最短路径长度。

Solution

由于最多10个宝箱，迷宫面积也很小，可以使用搜索。

dfs(id, step, cost) 表示当前处在编号id的宝箱处，已经走过了step个宝箱，花费了cost步
枚举下一个走到哪个宝箱，剪枝：如果当前花费步数已经超过之前的解法则剪枝。

```

#include <bits/stdc++.h>
using namespace std;
int x[11], y[11];
int ans = 1e9;
bool vis[11];
int n;
void dfs(int id, int step, int cost){
    if (step == n){
        ans = min(ans, cost + abs(x[id] - 10) + abs(y[id] - 1));
        return;
    }
    if (cost >= ans) return;
}

```

```

vis[id] = 1;
for (int i = 1; i <= n; i++)
    if (!vis[i]) dfs(i, step+1, cost+abs(x[id]-x[i])+abs(y[id]-y[i]));
vis[id] = 0;
}
int main(){
    cin >> n;
    x[0]=y[0]=1;
    for (int i = 1; i <= n; i++)
        cin >> x[i] >> y[i];
    dfs(0, 0, 0);
    cout << ans << endl;
    return 0;
}

```

D - 为美好的世界献上爆炎法术

考察点

凸包，旋转卡壳

题目大意

选取平面上 n 个点中的4个点，求构成的最大面积。旋转卡壳性质可得面积的单调性。
先求凸包（下面的是Andrew算法），求出凸包后枚举对角线的两个点，固定住后旋转卡壳求出最大面积。 n^2 解决。

Solution

```

#include<bits/stdc++.h>
#define ll long long
#define maxn 2005
using namespace std;
const double eps=10e-9;

inline int zt(double x){
    if(fabs(x)<eps) return 0;
    return x>0?1:-1;
}

struct node{
    double x,y;
    node operator -(const node &U)const{
        return (node){x-U.x,y-U.y};
    }
    node operator +(const node &U)const{
        return (node){x+U.x,y+U.y};
    }
    bool operator <(const node &U)const{
        return zt(x-U.x)?zt(x-U.x)<0:zt(y-U.y)<0;
    }
}a[maxn],hill[maxn];
int n,m,p[maxn];
double ans;

inline double xmul(node x,node y){

```

```

        return x.x*y.y-x.y*y.x;
    }
    inline bool equal(node x,node y){
        return (!zt(x.x-y.x)&&!zt(x.y-y.y));
    }
    inline void get_hill(){
        sort(a+1,a+n+1);
        int tot=0,now=0;
        for(int i=1;i<=n;i++){
            while(now>=2&&zt(Xmul(a[i]-a[p[now]],a[p[now]]-a[p[now-1]]))>=0) now--;
            p[++now]=i;
        }
        for(int i=1;i<=now;i++) hill[i]=a[p[i]];
        tot=now,now=0;
        for(int i=1;i<=n;i++){
            while(now>=2&&zt(Xmul(a[i]-a[p[now]],a[p[now]]-a[p[now-1]]))<=0) now--;
            p[++now]=i;
        }
        for(int i=now;i-->0; i--) if(!equal(a[p[i]],hill[tot])) hill[++tot]=a[p[i]];
        if(equal(hill[tot],hill[1])) tot--;
        n=tot;
    }

    inline int mo(int x,const int ha){
        if(x>=ha) return x-ha;
        else return x;
    }

    inline void solve(){
        for(int i=1;i<=n;i++){
            int pt1=i+1,pt2=i+3;
            for(int j=i+2;j<=n;j++){
                int topt=mo(pt1,n)+1;
                while(Xmul(hill[topt]-hill[i],hill[j]-hill[i])>Xmul(hill[pt1]-
hill[i],hill[j]-hill[i])){
                    pt1=topt;
                    topt=mo(pt1,n)+1;
                }

                topt=mo(pt2,n)+1;
                while(Xmul(hill[j]-hill[i],hill[topt]-hill[i])>Xmul(hill[j]-
hill[i],hill[pt2]-hill[i])){
                    pt2=topt;
                    topt=mo(pt2,n)+1;
                }

                ans=max(ans,(Xmul(hill[pt1]-hill[i],hill[j]-hill[i])+Xmul(hill[j]-
hill[i],hill[pt2]-hill[i]))/2.00);
            }
        }
    }

    int main(){
        scanf("%d",&n);
        for(int i=1;i<=n;i++) scanf("%lf%lf",&a[i].x,&a[i].y);
        get_hill();
        solve();
        printf("%.3lf\n",ans);
    }

```

```
    return 0;  
}
```