

1 pA

把黑色方塊全部移到最下面需要幾步

策略：記錄每排移了幾個答案直接加上要移動的距離（可能為負）

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 void Solve(){
5     int n, m;
6     scanf("%d%d", &n, &m);
7     int x[128] = {}, tmp, ans = 0;
8     for(int i = 0 ; i < n ; i++){
9         for(int j = 0 ; j < m ; j++){
10             scanf("%d", &tmp);
11             if(tmp){
12                 ans += (n - x[j] - 1) - i;
13                 x[j]++;
14             }
15         }
16     }
17     printf("%d\n", ans);
18 }
19
20 int main(){
21     int n;
22     scanf("%d", &n);
23     while(n--){
24         Solve();
25     }
```

2 pG

觀察可以發現 $f[n] = f[n-2] + f[n-3]$

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 typedef long long ll;
4
5 int main(){
6     int n, x;
7     ll ans[128];
8     ans[1] = ans[2] = ans[3] = 1;
9     for(int i = 4 ; i <= 100 ; i++)
10         ans[i] = ans[i-2] + ans[i-3];
11     scanf("%d", &n);
12     for(int i = 0 ; i < n ; i++){
13         scanf("%d", &x);
14         printf("%lld\n", ans[x]);
15     }
16     return 0;
17 }
```

3 pI

第一排給定 rank 對第二排做逆序數對

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define MAX 131072
4 typedef long long ll;
5
6 namespace BIT{
7     int v[MAX];
8     void init(){
9         memset(v, 0, sizeof(v));
10    }
11    void add(int x){
12        for( ; x < MAX ; x += x & -x)
13            v[x]++;
14    }
15    int query(int x){
16        int re = 0;
17        for( ; x > 0 ; x -= x & -x)
18            re += v[x];
19        return re;
20    }
21 };
22
23 int v[MAX];
24
25
26 void Solve(){
27     int n, x;
28     scanf("%d", &n);
29     unordered_map<int, int> u;
30     for(int i = 0 ; i < n ; i++){
31         scanf("%d", &x);
32         u[x] = i + 1;
33     }
34     ll ans = 0;
35     BIT::init();
36     for(int i = 0 ; i < n ; i++){
37         scanf("%d", &v[i]);
38     }
39     for(int i = n - 1 ; i >= 0 ; i--){
40         ans += BIT::query(u[v[i]]);
41         BIT::add(u[v[i]]);
42     }
43     printf("%lld\n", ans);
44 }
45
46
47 int main(){
48     int n;
49     scanf("%d", &n);
50     while(n--){
51         Solve();
52     }
53     return 0;
54 }
```

4 pJ

問你長度總合為 k 以內可以構造出幾個直角三角形

策略：從他給定的式子觀察可以發現 $a = 2xy, b = x^2 + y^2$ 我們只需要對 x, y 枚舉到 1000 就行了
接著做一次前綴和二分搜答案

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main(){
5     vector<int> ans;
6     for(int x = 1 ; x <= 1000 ; x++){
7         for(int y = 1 ; y < x ; y++){
8             if(__gcd(x, y) == 1){
9                 int a = 2 * x * y;
10                int b = x * x + y * y;
11                if(__gcd(a, b) == 1){
12                    ans.push_back(a+b);
13                }
14            }
15        }
16        sort(ans.begin(), ans.end());
17        for(int i = 1 ; i < (int)ans.size() ; i++){
18            ans[i] += ans[i-1];
19            if(ans[i] > 1000000){
20                ans.resize(i);
21                break;
22            }
23        }
24        int n, x;
25        scanf("%d", &n);
26        while(n--){
27            scanf("%d", &x);
28            printf("%d\n", int(upper_bound(ans.begin(), ans.end(), x) - ans.begin()));
29        }
30        return 0;
31 }
```

5 pK

給定兩排郵票規定不能剪相鄰的問最多可以拿到多少價值

策略：用動態規劃 $dp[h][w][3]$ 對於每個直列分三種狀態全都不拿拿上面跟拿下面

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define MAX 131072
4
5 int dp[MAX][3];
6 int v[MAX][2];
7
8 void Solve(){
9     int n;
10    scanf("%d", &n);
11    for(int i = 0 ; i < 2 ; i++)
12        for(int j = 0 ; j < n ; j++)
13            scanf("%d", &v[j][i]);
14    dp[0][0] = 0;
15    dp[0][1] = v[0][0];
16    dp[0][2] = v[0][1];
17    for(int i = 1 ; i < n ; i++){
18        dp[i][0] = max(dp[i-1][0], max(dp[i-1][1], dp[i-1][2]));
19        dp[i][1] = max(dp[i-1][0], dp[i-1][2]) + v[i][0];
20        dp[i][2] = max(dp[i-1][0], dp[i-1][1]) + v[i][1];
21    }
22    printf("%d\n", max(dp[n-1][0], max(dp[n-1][1], dp[n-1][2])));
23 }
24
25 int main(){
26     int n;
27     scanf("%d", &n);
28     for(int i = 0 ; i < n ; i++)
29         Solve();
30     return 0;
31 }
```

6 pL

給定 n 個人每個人都會連一個人問有多少人不會形成環

題解: topological sort 即可, 不在環上的點都會被拔掉

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define MAX 131072
4
5 int v[MAX], ans, deg[MAX];
6
7 void Solve(){
8     int n;
9     ans = 0;
10    scanf("%d", &n);
11    memset(deg, 0, sizeof(deg));
12    for(int i = 0 ; i < n ; i++){
13        scanf("%d", &v[i]);
14        v[i]--;
15        deg[v[i]]++;
16    }
17    for(int i = 0 ; i < n ; i++)
18        if(deg[i] == 0){
19            ans++;
20            int u = v[i];
21            while(1){
22                deg[u]--;
23                if(deg[u] == 0){
24                    deg[u] = -1;
25                    ans++;
26                    u = v[u];
27                } else {
28                    break;
29                }
30            }
31        }
32    printf("%d\n", ans);
33 }
34
35 int main(){
36     int n;
37     scanf("%d", &n);
38     while(n--){
39         Solve();
40     }
41 }
```