

提高组DAY1试题题解

破碎的饼干

题解

枚举每个碎片放在哪个位置，枚举左上角的格子是黑色还是白色，然后直接算出答案，取个min就好了。

标准代码

c++ 11

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

using namespace std;
typedef long long ll;
const int N = 210;
const int INF = 0x3f3f3f3f;
int _w;

int n, m, a[4][N][N], p[4], ans = INF, g[N][N];

void fill() {
    int k = p[0];
    for( int i = 0; i < n; ++i )
        for( int j = 0; j < n; ++j )
            g[i][j] = a[k][i][j];
    k = p[1];
    for( int i = 0; i < n; ++i )
        for( int j = 0; j < n; ++j )
            g[i+n][j] = a[k][i][j];
    k = p[2];
    for( int i = 0; i < n; ++i )
        for( int j = 0; j < n; ++j )
            g[i][j+n] = a[k][i][j];
    k = p[3];
    for( int i = 0; i < n; ++i )
        for( int j = 0; j < n; ++j )
            g[i+n][j+n] = a[k][i][j];
}

void check() {
    int now = 0, tmp = 0;
    for( int i = 0; i < m; ++i )
        for( int j = 0; j < m; ++j ) {
            int t = (i+j) & 1;
            if( g[i][j] != t ) ++now;
            else ++tmp;
        }
    ans = min( ans, now );
    ans = min( ans, tmp );
}
```

```

}

int main() {
    _w = scanf( "%d", &n );
    m = n*2;
    for( int k = 0; k < 4; ++k )
        for( int i = 0; i < n; ++i )
            for( int j = 0; j < n; ++j ) {
                char ch;
                _w = scanf( " %c", &ch );
                a[k][i][j] = ch - '0';
            }
    for( int i = 0; i < 4; ++i ) p[i] = i;
    do {
        fill(), check();
    } while( next_permutation(p, p+4) );
    printf( "%d\n", ans );
    return 0;
}

```

序列区间

题解

30pts : $n \leq 10$, $\max a \leq 10$ 。

数字很小，不会爆longlong，暴力枚举每个区间，暴力计算即可。

60pts : $n \leq 100$ 。

注意到所有数字的和最多也就是 $2e13$ 。

再乘k也不过 $2e18$ 。

所以依然可以暴力枚举每个区间。

在计算区间乘积的时候，如果乘积大于 $2e18$ ，那么这个区间肯定不合法。

解决了爆longlong的问题。

80pts : $\min a \geq 2$ 。

部分分提示正解。

就算区间里都是2，也很快就会爆longlong。

固定一个位置作为左端点，向右一个一个扫描，爆 $2e18$ 的时候停止就好了。

区间长度不会超过log级别，所以复杂度很低。

100pts。

有数字1的出现，意味着不能暴力了，因为区间可能很长。

其实对于连续的一段1稍微处理一下就没了。

(口胡)

相似题目：NOIp13火柴排队

标准代码

c++ 11

```
#include <cstring>
#include <algorithm>
#include <cstdio>

typedef long long ll;
const int N = 200010;
const ll INFLL = 3e18;
int _w;

int n, k, a[N];
ll ans;

int right[N];
void prelude() {
    for( int i = n; i >= 1; --i )
        if( a[i] == 1 )
            right[i] = a[i+1] == 1 ? right[i+1] : i;
}

bool check( ll a, ll b ) {
    return 1.0 * a * b <= INFLL;
}

ll solve( int L ) {
    ll p = 1, s = 0, ans = 0;
    for( int i = L; i <= n; ++i )
        if( a[i] == 1 ) {
            int R = right[i];
            int cnt = R-i+1;
            if( p % k == 0 ) {
                ll tmp = p/k-s;
                if( tmp >= 1 && tmp <= cnt )
                    ++ans;
            }
            i = R, s += cnt;
        } else {
            if( !check(p, a[i]) ) break;
            p *= a[i], s += a[i];
            if( p % k == 0 && p/k == s )
                ++ans;
        }
    return ans;
}

int main() {
    _w = scanf( "%d%d", &n, &k );
    for( int i = 1; i <= n; ++i )
        _w = scanf( "%d", a+i );
    prelude();
    for( int i = 1; i <= n; ++i )
        ans += solve(i);
    printf( "%lld\n", ans );
    return 0;
}
```

```
}
```

友谊序列

题解

70pts : $n \leq 2000$ 。

数字很小，开个数组记录。

然后直接模拟，送分一步到位。

90pts : $v \leq 1e6$ 。

学过生成函数和多项式你就知道，这个东西在数字很小的时候直接用FWT做就没了。

100pts : 输出样例即可获得满分。

假如有一对 (i, j) 满足 $a[i] \text{ xor } b[j] = c$ 且 c 在集合 S 中。

当 c 在数组 a 中，有 $a[i] \text{ xor } b[j] = a[k]$ ，且 $a[k] \text{ xor } b[j] = a[i]$ 。

当 c 在数组 b 中，有 $a[i] \text{ xor } b[j] = b[k]$ ，且 $a[i] \text{ xor } b[k] = b[j]$ 。

这样的东西总是成对出现的。

所以答案一定是偶数。

相似题目：NOIp17小凯的疑惑

标准代码

c++ 11

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

using namespace std;
int _w;

int main() {
    int T;
    _w = scanf( "%d", &T );
    while( T-- ) puts("0!");
    return 0;
}
```