

= Codeforces Round #664 (Div. 2) =

**A. [<https://codeforces.com/contest/1395/problem/A> Boboniu Likes to Color Balls]**

构成回文串的充要条件是，出现奇数次的字符最多有1种。

进行操作后会使得rgbw四种字符的奇偶性改变。

分别根据进行操作/不进行操作判断

```
<source lang="cpp">#include <bits/stdc++.h>
```

```
using namespace std;
```

```
typedef long long LL;
```

```
int main() {
```

```
    int t;
```

```
    cin >> t;
```

```
    while (t--) {
```

```
        LL r, g, b, w;
```

```
        cin >> r >> g >> b >> w;
```

```
        if ((r == g) && (r == b)) {
```

```
            cout << "Yes" << endl;
```

```
        } else {
```

```
            int c1 = (r & 1) + (g & 1) + (b & 1);
```

```
            int c2 = w & 1;
```

```
            int cnte = c1 + (w & 1);
```

```
            int cnt2 = 3 - c1 + 1 - (w & 1);
```

```
            if (cnte == 1 || cnte == 0 || (r > 0 && g > 0 && b > 0 && (cnt2 == 1 || cnt2 == 0))) {
```

```
                cout << "Yes" << endl;
```

```
            }
```

```
            else cout << "No" << endl;
```

```
}
```

```
}
```

```
}</source>
```

## B. [<https://codeforces.com/contest/1395/problem/B> Boboniu Plays Chess]

这里给出的覆盖策略是先走完当前行，再按顺序走完1-n行，跳过已走过的值。

```
<source lang="cpp">#include <bits/stdc++.h>
```

```
using namespace std;
```

```
typedef long long LL;
```

```
int p[200][200] = {};
```

```
int pr(int x, int y){
```

```
    p[x][y] = 1;
```

```
    printf("%d %d\n", x, y);
```

```
}
```

```
int main() {
```

```
    int n, m, x, y;
```

```
    cin>>n>>m>>x>>y;
```

```
    p[x][y] = 1;
```

```
    pr(x, y);
```

```
    int xx = x;
```

```
    if(y > 1){
```

```
        y = 1;
```

```
        pr(x, y);
```

```
    }
```

```
    int y1 = y;
```

```
for (y = 1; y <= m; ++y) {  
    if(!p[x][y]){  
        y1 = y;  
        pr(x, y);  
    }  
}  
  
y = y1;  
  
for (int i = 1; i <= n; ++i) {  
    if(i == xx){  
        continue;  
    }  
  
    x = i;  
    pr(x, y);  
  
    if(y > 1){  
        y = 1;  
        pr(x, y);  
    }  
  
    y1 = y;  
  
    for (y = 1; y <= m; ++y) {  
        if(!p[x][y]){  
            y1 = y;  
            pr(x, y);  
        }  
    }  
}  
  
y = y1;
```

```
}
```

```
}</source>
```

### C. [<https://codeforces.com/contest/1395/problem/C> Boboniu and Bit Operations]

贪心，使高位尽可能多地为0。从高位到低位枚举。如果结果中一位可以得到0，则清除所有使这一位得不到0的选项。

```
<source lang="cpp">#include <bits/stdc++.h>
```

```
using namespace std;
```

```
typedef long long LL;
```

```
int main() {
```

```
    int n, m;
```

```
    int a[300], b[300], mi[300];
```

```
    cin>>n>>m;
```

```
    for (int i = 0; i < n; ++i) {
```

```
        scanf("%d", a+i);
```

```
    }
```

```
    set<int> poss[300] = {};
```

```
    for (int i = 0; i < m; ++i) {
```

```
        scanf("%d", b+i);
```

```
        for (int j = 0; j < n; ++j) {
```

```
            poss[j].insert(b[i]);
```

```
        }
```

```
    }
```

```
    int ans = 0;
```

```
    for (int k = 8; k >= 0; --k) {
```

```
int kk = 1 << k;

bool okA = true;

for (int i = 0; i < n; ++i) {

    bool ok = false;

    for(int p : poss[i]){

        if((p & a[i] & kk) == 0){

            ok = true;

            break;

        }

    }

    if(!ok){

        okA = false;

        break;

    }

}

if(!okA){

    ans |= kk;

}

else{

    for (int i = 0; i < n; ++i) {

        for(auto it = poss[i].begin(); it != poss[i].end(); ){

            if(*it & a[i] & kk){

                poss[i].erase(it++);

            }else it++;

        }

    }

}
```

```
    }  
}  
}  
cout<<ans<<endl;  
}</source>
```

#### D. [<https://codeforces.com/contest/1395/problem/D> Boboniu Chats with Du]

这道题不同情况下，最优解中被禁言的次数不一定...

因此需要枚举被禁言次数的可能情况。

在禁言次数一定时，取尽可能大的值放入未被禁言的部分中，求前缀和后可以 $O(1)$ 得到结果。

```
<source lang="cpp">#include <bits/stdc++.h>  
using namespace std;  
typedef long long LL;  
int a[100006];  
LL sum[100006] = {};  
int main() {  
    int n, m, d;  
    cin >> n >> d >> m;  
    for (int i = 0; i < n; ++i) {  
        scanf("%d", a + i);  
    }  
    sort(a, a + n);  
    int cnt1 = 0;  
    for (int j = 0; j < n; ++j) {  
        sum[j+1] = sum[j] + a[j];  
    }  
}
```

```
        if (a[j] > m) cnt1++;
    }

    LL ans = 0;

    for (int i = 0; n >= (i - 1) * (d + 1) + 1; ++i) {

        if (cnt1 < i) continue;

        if (cnt1 > i * (d + 1)) continue;

        int extra = i == 0 ? n : n - (i - 1) * (d + 1) - 1;

        int s = n - cnt1;

        ans = max(ans, sum[n] - sum[n - i] + sum[s] - sum[max(0, s - extra)]);

    }

    cout<<ans<<endl;
}
</source>
```

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:too\\_low:cf1395\\_hj&rev=1597394684](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:too_low:cf1395_hj&rev=1597394684)



Last update: **2020/08/14 16:44**