

A.LCM Problem

<https://codeforces.com/contest/1389/problem/A>

题意：找到 $[l, r]$ 范围内的两个数 $x < y$ 使得其最小公倍数也在 $[l, r]$ 范围内。

设 $x = p * \gcd(x, y)$ $y = q * \gcd(x, y)$ p, q 互质 $\text{lcm}(x, y) = pq * \gcd(x, y)$ x 确定时 $y = 2x$ 时 $\text{lcm}(x, y) = y$ 取到最小值

$x = l$ 时存在解有 $= 2 * l$ 判断 r 是否小于 $2 * l$ 即可。

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

using namespace std;
typedef long long LL;

int main() {
    int t = 0;
    cin >> t;
    while (t--) {
        LL l, r;
        cin >> l >> r;
        if (r < 2 * l) cout << -1 << ' ' << -1 << endl;
        else cout << l << ' ' << 2 * l << endl;
    }
}
```

B.Array Walk

题意：给定一个数组，起始位置在下标1处，可以选择向左走与向右走，不可重复向左走，向左走总次数不得超过 p ，总共走 k 次，求经过路径上数组元素值和的最大值。

贪心。对于 $1 \rightarrow x$ 范围内的路径，三种情况可能为最优值：

找到相邻两项和的最大值，在这相邻两项间重复走 $\min(p, k - p/2)$ 次。

找到相邻两项和的最大值，在这相邻两项间重复走 $\min(p, k - p/2)$ 次，再在结尾向左走1次。

$1 \rightarrow x \rightarrow x - 1$

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

using namespace std;
typedef long long LL;
```

```
LL a[100005];
LL maxA2[100005];
LL tot[100005];
int main() {
    int t = 0;
    cin >> t;
    while (t--) {
        int n, k, z;
        LL ans = 0;
        cin >> n >> k >> z;
        LL tt = 0;
        for (int i = 0; i < n; ++i) {
            scanf("%d", a+i);
            if(i) tt += a[i];
            tot[i] = tt;
        }
        LL a2 = 0;
        for (int i = 0; i < n - 1; ++i) {
            a2 = max(a2, a[i] + a[i+1]);
            maxA2[i] = a2;
        }
        ans = tot[k];
        for (int i = 1; i <= k; ++i) {
            int rep = min(z, (k - i) / 2);
            if(k == i + rep * 2) ans = max(ans, maxA2[i - 1] * rep +
tot[i]);
            if(k == i + 1 && z) ans = max(ans, tot[i] + a[i-1]);
            if(k == i + 1 + rep * 2 && z >= rep + 1) ans = max(ans, maxA2[i
- 1] * rep + tot[i] + a[i-1]);
        }
        cout << ans + a[0] << endl;
    }
}
```

C.Good String

题意：长度为n的good string是 $s[i] = s[(i+2) \% n]$ 的字符串

给定一个0-9构成的字符串，求使其变为good string需要删除的最少字符数

直接暴力枚举，在0-9中选2个数字构成good string[]计算每次选择需要删除的字符数，找到最小的答案。

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

using namespace std;
typedef long long LL;
```

```

int main() {
    int t = 0;
    cin >> t;
    while (t--) {
        string s;
        cin>>s;
        int l = s.length();
        int ans = INT32_MAX;
        for (int i = '0'; i <= '9'; ++i) {
            for (int j = '0'; j <= '9'; ++j) {
                int e = 0;
                bool rev = 0;
                for (int k = 0; k < l && e < ans; ++k) {
                    char c = rev ? j : i;
                    if(s[k] != c) e++;
                    //else if(k == l - 1 && i != j && (l & 1))e++;
                    else rev = !rev;
                }
                if(rev && (i != j))e++;
                ans = min(ans, e);
            }
        }
        cout<<ans<<endl;
    }
}

```

D.Segment Intersections

题意：给定2种区间，每种区间有n个，2种不同区间为一组。每次操作可以将区间向左/右扩展一个单位。求使各组区间重叠长度达到k的最小步数。

最优情况下，扩展时有3个过程：不相邻→相邻→区间完全重叠→继续扩展

三个过程的重叠数收入分别为0，扩展数、扩展数/2

记录每个过程的扩展长度，枚举选i组区间扩展到重叠k次需要的扩展数，取最小值。

需要注意已经重叠的情况

```

#include <bits/stdc++.h>

using namespace std;
typedef long long LL;

int main() {
    int t = 0;
    cin >> t;
    while (t--) {

```

```
LL n, k;
cin>>n>>k;
int l1, r1, l2, r2;
cin>>l1>>r1>>l2>>r2;
LL pre = max(0, max(l1, l2) - min(r1, r2));
LL most = max(r1, r2) - min(l1, l2);
LL ist = max(0, min(r1, r2) - max(l1, l2)) * (LL)n ;
k -= ist;
if(k <= 0){
    cout<<0<<endl;
    continue;
}
LL ans = INT32_MAX;
for (int i = 1; i <= n; ++i) {
    LL cur = pre * i;
    cur += min((LL)k, most * i);
    if(k > most * i){
        cur += 2 * (k - most * i);
    }
    ans = min(ans, cur);
}
cout<<ans<<endl;
}
```

From:

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

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:too_low:cfedu92hj



Last update: **2020/07/31 17:37**