

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 $1-x$ 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

<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。

D.Segment Intersections

<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。

E.Segment Intersections

From:

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

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:too_low:cfedu92hj&rev=1596187219 

Last update: **2020/07/31 17:20**