

AtCoder Beginner Contest 174

A-Air Conditioner

水题，水到直接在网页的提交代码框里写了。

略

B Distance

题意：有多少个点距离原点小于 d

```
#include <algorithm>
#include <cmath>
#include <cstdio>
#include <cstring>
#include <iostream>
#include <string>
using namespace std;
typedef long long ll;
int main()
{
    ll n,d;
    cin >> n >> d;
    d *= d;
    int ans = 0;
    for (int i = 1; i <= n;i++)
    {
        ll x, y;
        cin >> x >> y;
        if (x*x + y*y <= d)
            ans++;
    }
    cout << ans << endl;
    return 0;
}
```

C Repsept

题意：给出一个正整数 K 问在全部由7组成的数字中，求出一个最短的数字可以被 K 整除，如果没有就输出-1。

思路 $K \leq 1e6$,所以直接暴力计算模数即可，同时记录模数，出现循环，就不存在，输出-1.

```
#include <algorithm>
#include <cmath>
#include <cstdio>
#include <cstring>
#include <iostream>
#include <string>
#include <map>
using namespace std;
typedef long long ll;
int vztd[5000000];
int main()
{
    ll k;
    cin >> k;
    ll tem = 7;
    ll ans = 1;
    while (tem < k)
        tem = tem * 10 + 7, ans++;
    for (;;)
    {
        if (tem % k == 0)
        {
            cout << ans << endl;
            return 0;
        }
        ans++;
        tem = tem * 10 + 7;
        tem %= k;
        if (vztd[tem] == 1)
            break;
        else
            vztd[tem] = 1;
    }
    cout << -1 << endl;
    return 0;
}
```

D Alter Altar

水题

暴力前后枚举，遇到相反就交换。

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

```

#include <string>
#include <map>
using namespace std;
typedef long long ll;
char s[200009];
int main()
{
    int n;
    cin >> n;
    scanf("%s", s+1);
    int ans = 0, l = 1, r = n;
    while (l<r)
    {
        while (s[l] == 'R')
            l++;
        while (s[r] == 'W')
            r--;
        if (l >= r)
            break;
        ans++;
        swap(s[l], s[r]);
    }
    cout << ans << endl;
    return 0;
}

```

E Logs

题意：给出一堆原木，每次可以挑一根砍成两段，问砍k次，之后要求最长的原木的最小值。

题解：看到这个最大值最小就应该想到二分答案，直接二分即可，注意round up 是向上取整，，多个翻译软件都会翻译成四舍五入。

```

#include <algorithm>
#include <cmath>
#include <cstdio>
#include <cstring>
#include <iostream>
#include <string>
#include <map>
using namespace std;
const int MAX = 2e5 + 20;
typedef long long ll;
int pic[MAX], n, k;
bool check(int goal)
{
    int tem = k, i;
    for (i = 1; i <= n; i++)

```

```
{
    if (pic[i] <= goal)
        continue;
    else
    {
        int ans = 0, logg = pic[i];
        ans = logg / goal;
        if (logg % goal == 0)
            ans--;
        tem -= ans;
        if (tem < 0)
            return false;
    }
}
return true;
}
int main()
{
    cin >> n >> k;
    int l = 1, r = 0;
    for (int i = 1; i <= n; i++)
    {
        scanf("%d", &pic[i]);
        r = max(pic[i], r);
    }
    while (l + 2 < r)
    {
        int mid = (l + r) >> 1;
        if (check(mid))
            r = mid;
        else
            l = mid + 1;
    }
    for (int i = l; i <= r; i++)
        if (check(i))
        {
            cout << i << endl;
            break;
        }
    return 0;
}
```

F Range Set Query

题意：静态区间求种类数。经典题了

题解：树状数组离线，主席树在线都可以水过。

```
#include <algorithm>
#include <iostream>
#include <map>
using namespace std;
struct data
{
    int z, y, xh;
    bool operator<(const data &x) const
    {
        if (y < x.y)
            return 1;
        else
            return 0;
    }
} s[501001];
int c[500011];
int a[500101];
int h[500101];

map<int, int> Map;
int n;
int lowbit(int i)
{
    return i & (-i);
}
void add(int i, int x)
{
    while (i <= n)
    {
        c[i] += x;
        i += lowbit(i);
    }
}
int sum(int x)
{
    int ans = 0;
    while (x > 0)
    {
        ans += c[x];
        x -= lowbit(x);
    }
    return ans;
}
int main()
{
    ios::sync_with_stdio(0);
    int m;
    cin >> n >> m;
    for (int i = 1; i <= n; ++i)
        cin >> a[i];
```

```
for (int i = 1; i <= m; ++i)
{
    cin >> s[i].z >> s[i].y;
    s[i].xh = i;
}

sort(s + 1, s + 1 + m);
int last = 0;
for (int i = 1; i <= m; ++i)
{
    ++last;
    for (int j = last; j <= s[i].y; ++j)
    {
        if (Map[a[j]])
        {
            add(Map[a[j]], -1);
            Map[a[j]] = j;
            add(Map[a[j]], 1);
        }
        else
        {
            Map[a[j]] = j;
            add(j, 1);
        }
    }
    h[s[i].xh] = sum(s[i].y) - sum(s[i].z - 1);
    last = s[i].y;
}
for (int i = 1; i <= m; ++i)
    cout << h[i] << endl;
}
```

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

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:too_low:abc174_cy



Last update: **2020/08/07 17:20**