

[编程]10分] 求最少操作数

时间限制: C/C++ 5秒, 其他语言 10秒

空间限制: C/C++ 65536K, 其他语言 131072K

64bit IO Format: %lld

语言限定: C++11(clang++ 3.9), C++14(g++5.4), C(clang 3.9),  
Python(2.7.3), Python3(3.5.2)

本题可使用本地IDE编码, 不做跳出限制, 编码后请点击“保存并调试”按钮  
进行代码提交。

■ 题目描述

给定 $x$ 一个初始正整数值 $m$ 和三种操作“+1”, “/2”, “/3”。每次可以在上述三种操作中选择一种作用在 $x$ 上(除法是直接除, 可能会有小数, 而不是取除完以后的商), 请问最少需要多少次操作能使得 $x$ 的值变为1。

输入描述:

第一行输入为 $x$ 的初始值 $m$ , 范围 $0 <= m < 1000000$ 。

输出描述:

第一行输出一个整数 $t$ 表示最少的操作数。

**示例1** 输入输出示例仅供调试, 后台判题数据一般不包含示例

输入

复制

1

输出

复制

输入

复制

1

输出

复制

0

**示例2** 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

0

输出

复制

1

**示例3** 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

10

输出

复制

4

```
#include<stdio.h>
#include<algorithm>
#include<string>
#include<iostream>
#include<string.h>
#include<queue>
using namespace std;

int f(int n)
{
    if(n == 1)
        return 0;
    else if( n <= 3)
        return 1;

    int val1 = n, val2 = n;
    if(n % 2 == 0)
        val1 = min(val1, f(n/2) + 1);
    else
        val1 = min(val1, f((n+1)/2) + 2);

    if(n % 3 == 0)
```

```

        val2 = min(val2, f(n/3) + 1);
    else if(n % 3 == 2)
        val2 = min(val2, f((n+1)/3) + 2);
    else
        val2 = min(val2, f((n+2)/3) + 3);

    return min(val1, val2);
}

int main()
{

    int n;
    cin >> n;

    cout << f(n) << endl;
}

/*
int main()
{

    int n;
    cin >> n;

    if(n == 1)
        cout << 0 << endl;
    else if( n <= 3)
        cout << 1 << endl;
    else
    {
        int count = 0;
        while(n > 3)
        {
            if(n % 3 == 0)
                n /= 3;
            else if (n % 2 == 0)
                n /= 2;
            else
                n += 1;
            //cout << n << endl;
            count++;
        }

        if(n == 1)
            cout << count << endl;
        else if( n <= 3)
            cout << count + 1 << endl;

    }

    return 0;
}
*/

```

时间限制: C/C++ 1秒, 其他语言 2秒

空间限制: C/C++ 131072K, 其他语言 262144K

64bit IO Format: %lld

语言限定: C++11(clang++ 3.9), C++14(g++5.4), C(clang 3.9),

Python(2.7.3), Python3(3.5.2), pypy2(pypy2.7.13), pypy3(pypy3.6.1)

本题可使用本地IDE编码, 不做跳出限制, 编码后请点击“保存并调试”按钮进行代码提交。

#### ■ 题目描述

*特别注意, 对于这个题目来说, Python的时间限制是10s。使用Python可以通过所有数据 (若时间复杂度是对的而且写的较好将可以通过全部数据), 由于其运算过慢, 建议使用c++。*

*由于Python时间限制较大, 使用Python的话, 判题会特别慢 (可能最多会等5分钟甚至以上)。大家在等待时可以先想其他题。*

*pypy比普通python要快一些, 使用python与题的朋友可以交对应pypy, 交python会比较慢, 但是也可以通过所有数据。*

小A有 $n$ 个糖果, 这天, 他的 $m$ 个朋友将要来拜访, 小A将自己所有的糖果摆成一排, 编号为  $1, 2, \dots, n$ , 其对应的好吃程度为  $a_1, \dots, a_n$ 。每个朋友来后, 会从区间  $[l_x, r_x]$  的糖果中依次选择好吃程度最小的  $k$  个糖果吃掉 (因为不好意思吃太好吃的)。若这个区间糖果数量比  $k$  少, 就把区间内所有糖果全吃掉。若区间内没有糖果, 则输出-1。

*也就是说, 问题是这样的, 给定一个数列, 多次询问, 每次询问其中的一个区间, 将其中最小的  $k$  个数字一个一个去掉 (若区间内的数字个数小于 $k$ , 就全去掉。), 并请输出去掉的最后一个数字是多少。若区间内没有数字, 则输出-1。*

此题中, 所有数据均为随机生成。

#### 输入描述:

第一行两个整数 $n, m$ ，代表总糖果个数是 $n$ ，朋友个数是 $m$   
接下来 $n$ 个整数 $a_1 \dots a_n$ ，代表这 $n$ 个糖果的对应好吃程度。  
接下来 $m$ 行，每行三个整数， $l, r, k$ ，分别代表区间的范围和选择的糖果个数。

#### 输出描述:

输出 $m$ 个数字，代表这个朋友吃掉的最后一个糖果是哪个

**示例1** 输入输出示例仅供调试，后台判题数据一般不包含示例

#### 输入

[复制](#)

```
4 3
1 2 3 4
1 4 2
1 3 2
1 2 1
```

#### 输出

[复制](#)

```
2
3
-1
```

#### 说明

询问1中，吃掉了1 2两个糖果  
询问2中，只吃了3一个糖果  
询问3中，没有吃糖果

#### 备注:

**特别注意，对于这个题目来说，Python的时间限制是10s。使用Python可以通过所有数据（若时间复杂度是对的而且写的较好可以通过全部数据），由于其运算过慢，建议使用C++。**  
**由于Python时间限制较大，使用Python的话，判题会特别慢（可能最多会等5分钟甚至以上）。大家在等待时可以先想其他题。pypy比普通python要快一些，使用python写题的朋友可以交对应pypy，交python会比较慢，但是也可以通过所有数据。**

10%的数据满足 $n \leq 100, m \leq 10$   
20%的数据满足 $n \leq 1000, m \leq 20$   
20%的数据满足 $n \leq 10000, m \leq 20$   
50%的数据满足 $n \leq 1000000, m \leq 40$   
除此之外，所有数据满足 $1 \leq l \leq r \leq n, 1 \leq k \leq n, 1 \leq k \leq 10^9$ ，保证所有数据均为随机生成。

```
#include<stdio.h>
#include<algorithm>
#include<string>
```

```

#include<iostream>
#include<string.h>
#include<queue>
using namespace std;

const int INF = 0x3f3f3f3f;
int num[1000005];
int visit[1000005];
int main()
{
    int n, m;
    cin >> n >> m;

    for(int i = 1; i <= n; i++)
        cin >> num[i];

    priority_queue <int, vector<int>, greater<int> > que;
    memset(visit, 0x3f, sizeof(visit));
    int l, r, k;
    for(int i=0; i<m; i++)
    {
        cin >> l >> r >> k;

        for(int j=l; j<=r; j++)
        {
            if(visit[j] == INF)
                que.push(num[j]);
        }

        if(que.empty())
            cout << -1 << endl;
        else if(que.size() >= k)
        {
            int x = k;
            int val;
            while(x--)
            {
                val = que.top();
                que.pop();

                for(int t=l; t<=r; t++)
                {
                    if(num[t] == val && visit[t] == INF)
                    {
                        visit[t] = 0;
                    }
                }
            }
            cout << val << endl;
        }
        else
        {
            int val;
            while(que.size())
            {

```

```
        val = que.top();
        que.pop();
    }

    cout << val << endl;

    for(int t=1; t<=r; t++)
        visit[t] = 0;
}

while(!que.empty())
    que.pop();

}

return 0;
}
/*
4 3
1 2 3 4
1 4 2
1 3 2
1 2 1
*/
```

[编程]30分] 吃饼干

时间限制: C/C++ 2秒, 其他语言 4秒

空间限制: C/C++ 65536K, 其他语言 131072K

64bit IO Format: %lld

语言限定: C++11(clang++ 3.9), C++14(g++5.4), C(clang 3.9),

Python(2.7.3), Python3(3.5.2), pypy2(pypy2.7.13), pypy3(pypy3.6.1)

本题可使用本地IDE编码, 不做跳出限制, 编码后请点击“保存并调试”按钮进行代码提交。

#### ■ 题目描述

**特别注意, 对于这个题目来说, Python的时间限制是10s。使用Python可以基本通过所有数据, 由于其运算过慢, 建议使用c++。**

**由于Python时间限制较大, 使用Python的话, 判题会特别慢 (可能最多会等5分钟甚至以上)。大家在等待时可以先想其他题。**

**pypy比普通python要快一些, 使用python与题的朋友可以交对应pypy, 交python会慢**

小A有  $k-1$  个朋友。有一天, 亲戚家托他的朋友给他一块长为  $n$ , 宽为  $m$  的正方形饼干, 需要通过  $k-1$  个朋友一个一个传下去最后交给他。由于小A的朋友喜欢“贪污”一部分饼干, 每次传递时, 这个人会产生传递来的饼干面积的“高兴值”, 在传递过程中, 他的每个朋友都会从获得的饼干中取出一个子矩形 (子矩形不包含自身, 行和列都得小于刚开始的值), 并把除了这个部分之外的部分全都吃掉, 并将剩下的部分交给下一个人, 需要注意, 最终交给小A, 小A也会产生高兴值。请问对于所有的可能情况中, 高兴值积之和是多大呢?

换句话说, 题意如下:

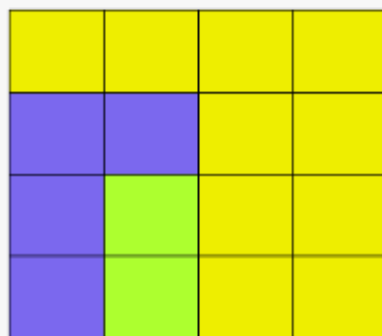
有一个大矩形, 我们要经过  $k$  个人之手, 共  $k-1$  次从大矩形中选择一个子矩形, 请问对于所有选择的情况来说, 这  $k$  次的矩形面积之积的和是多大呢?



比如一个3\*3的矩形，可以从中选择4种2\*2的矩形（因为横着有两种选法，纵着也有两种选法），这4种方案要分开计算，然后求和。也就是若询问为3\*3的矩形，选两次，从3\*3选2\*2会让答案加上 $(3*3)*(2*2)*4$ 。

**也就是说，问题是这样的，从一个大矩形每次选一个长和宽都小于它的子矩形，然后求选了k次以后（k次包括初始矩形），选的这k个矩形面积之积，然后对所有选择的情况的积求和**

为了让题意更明确，消除歧义：



如图所示的图案，高兴值之积为 $4*4$ (最大的矩形)\* $2*3$ (第二大的矩形)\* $1*2$ (最小的矩形)=192

只考虑最小的两个矩形的话（若形状固定），最里面的矩形有 $2*2=4$ 种选择方案，也就是说，如果固定了最大俩矩形的相对位置，答案应该加上 $192*4=768$ 。

从上面的描述可以看出，答案会非常大。请输出答案模998244353的值。也就是说，如果答案是998244354，请输出1。计算时注意越界哦~

输入描述:

共三个数字 $n$ 、 $m$ 、 $k$ ，分别代表饼干的长宽和小 A 加上朋友的总人数

输出描述:

各测试用例在1行内输出一个整数，代表这次询问的答案。

**示例1** 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

4 4 2

输出

复制

4096

说明

答案为 $4*4*$

$(1*1*4*4+1*2*4*3+1*3*4*2+...+3*3*2*2)=4096$

示例2 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
10 10 1
```

输出

复制

```
100
```

说明

```
答案为10*10=100
```

备注:

特别注意，对于这个题目来说，Python的时间限制是10s。使用Python可以基本通过所有数据，由于其运算过慢，建议使用C++。

由于Python时间限制较大，使用Python的话，判题会特别慢（可能最多会等5分钟甚至以上）。大家在等待时可以先想其他题。pypy比普通python要快一些，使用python写题的朋友可以交对应pypy，交python会慢

30%的数据满足 $n, m \leq 20, k \leq \min(n, m)$

20%的数据满足 $n, m \leq 50, k \leq \min(n, m)$

30%的数据满足 $n, m \leq 500, k \leq \min(n, m)$

20%的数据满足 $n, m \leq 10^5, k \leq \min(n, m, 500)$