2022 新春公开赛题解

正字

https://www.luogu.com.cn/problem/U192216

题目描述

Alice计数时喜欢画“正”字，例如 $12 = 2\times$ 正 $+2, 111 = 22\times$ 正 $+1$。

而Bob只了解二进制，例如 $12 = 1100, 111 = 1101111$

Alice进行了一次计数，结果为： $x\times$ 正 $+y$，请你帮助她把这个结果转化成二进制告诉Bob。

输入格式

第一行1个整数 q，代表有 q 组询问

接下来每行2个整数 x, y，代表一次 Alice 的计数结果

输出格式

输出 q 行，每行输出一个二进制数字代表Alice的计数结果

输入输出样例

输入 #1

```
5
2 2
22 1
0 0
3436 4
1000000000 4
```

输出 #1

```
1100
1101111
0
100001100100000
1001010100000001011111001000000100
```

说明/提示

对于所有测试点：$1 \le q \le 10^3, 0 \le x \le 10^9, 0 \le y < 5$

子任务1（30分）：满足 $x = 0$

子任务2（20分）：满足 $x \le 100$

子任务3（30分）：满足 $x \le 10^8$

子任务4（20分）：满足 $x \le 10^9$

本题主要考察进制转换。

将 `5x+y` 转换成二进制输出即可，别忘了开`long long`、以及特判0。

复杂度 O(q*log(5e9))。

```
#include<bits/stdc++.h>
#define MAXN 100005
using namespace std;
typedef long long ll;

void work(ll x){
    if(x==0){
        cout<<"0"<<endl;
        return;
    }
    int b[101], top = 0;
    while(x){
        b[++top] = x%2;
        x /= 2;
    }
    for(int i=top;i>=1;i--){
        cout<<b[i];
    }
    cout<<endl;
}
```

```
int main(){
    int q;
    cin>>q;
    ll x,y;
    while(q--){
        cin>>x>>y;
        work(5*x+y);
    }
    return 0;
}
```

统计

https://www.luogu.com.cn/problem/U192219

题目描述

一个长度为 n 的字符串 s，字符串中仅包含小写字母 a, b。定义子串 $s(l, r)$ 表示由 s 中从左往右数，第 l 到第 r 个字符依次连接形成的字符串。请你找出所有符合以下条件的子串 $s(l, r)$：

- $l \leq r$
- 子串中 a 的数量恰好为 2 的倍数
- 子串中 b 的数量恰好为 3 的倍数
- 子串中 a, b 的数量之差的绝对值恰好为 5 的倍数

你需要输出的是所有满足条件子串的长度之和

输入格式

第一行1个整数 n，代表字符串长度。

第二行1个字符串 s，保证仅包含小写字母 a, b。

输出格式

输出一个整数，代表所有满足条件子串的长度之和

输入输出样例

输入 #1

```
12
abaaaaabaaba
```

输出 #1

```
22
```

输入 #2

```
16
aabbbbaabbbbabbb
```

输出 #2

```
13
```

输入 #3

```
20
aabbbbbabbbbaaaabaaa
```

输出 #3

```
51
```

输入 #4

```
24
bbabaaaaabbbbbabaabbbbaa
```

输出 #4

```
110
```

输入 #5

```
16
aaaaaaaaaaaaaaaa
```

输出 #5

```
70
```

说明/提示

【样例解释】

对于样例1，满足条件的子串有 $s(1,11)$ 和 $s(2,12)$，它们都包含 8 个 a、3个 b，a,b 数量之差为 5。

对于样例2，符合条件的子串只有 $s(2,14)$。

【数据范围】

对于所有测试点：$1 \le n \le 10^5$，字符串 s 中仅包含小写字母 a,b。

子任务1（30分）：$n \le 100$

子任务2（20分）：$n \le 2000$

子任务3（15分）：$n \le 10^5$，字符串 s 中没有 b

子任务4（35分）：$n \le 10^5$

本题主要考察前缀和统计思想。

枚举子串结尾位置，设此时 a,b 数量分别为 na, nb。可以按 na-nb 的符号分成2类，各自记录na模2、nb模3、abs(na-nb)模5的：

1. 前缀数量 `cnt[0/1][na%2][nb%3][abs(na-nb)%5]`

2. 前缀总长度 `len[0/1][na%2][nb%3][abs(na-nb)%5]`

若 na>=nb，设 z = (na-nb)%5，则其和 na>=nb 中的那些相同z可以构成答案（由于 (z-z)%5 = 0），另外可以和 na<nb 中的 z-5 可以构成答案（由于 (z-(z-5))%5 = 0)。na < nb 时同理。

边枚举结尾位置边统计 cnt, len 2个数组，总复杂度线性。

```
#include<bits/stdc++.h>
#define MAXN 100005
using namespace std;
typedef long long ll;
int n;
char s[MAXN];
int cnt[2][2][3][5];
ll len[2][2][3][5];

int main(){
    cin>>n;
    cin>>s+1;

    cnt[0][0][0][0] = 1;
    ll ans = 0;
```

```
    int na = 0, nb = 0;
    for(int i=1;i<=n;i++){
        if(s[i]=='a') ++na;
        else ++nb;
        int x = na%2, y = nb%3, z1 = abs(na-
nb)%5, z2 = (5-abs(na-nb)%5)%5;
        if(na >= nb){
            ans += (ll)cnt[0][x][y][z1]*i -
len[0][x][y][z1];
            ans += (ll)cnt[1][x][y][z2]*i -
len[1][x][y][z2];
            cnt[0][x][y][z1]++;
            len[0][x][y][z1] += i;
        }
        else{
            ans += (ll)cnt[1][x][y][z1]*i -
len[1][x][y][z1];
            ans += (ll)cnt[0][x][y][z2]*i -
len[0][x][y][z2];
            cnt[1][x][y][z1]++;
            len[1][x][y][z1] += i;
        }
    }
    cout<<ans<<endl;
    return 0;
}
```

决断

https://www.luogu.com.cn/problem/U192221

题目描述

给定一个长度为 n 的数组 $a[1...n]$，你需要找到一个相同长度的数组 $b[1...n]$，满足：

- $b[1]=0$
- 对于 $i=2...n$，满足$0\le b[i]\le b[i-1]+1$
- $ans=\sum a[i]\times b[i]=a[1]\times b[1]+a[2]\times b[2]+...+a[n]\times b[n]$ 的值最大

请你输出这个最大的 ans 值

输入格式

第一行1个整数 n

第二行 n 个整数 $a[1...n]$

输出格式

输出一个整数，代表 ans 的最大值

输入输出样例

输入 #1

```
5
1 0 -6 -2 4
```

输出 #1

```
6
```

输入 #2

```
5
0 -2 0 -5 4
```

输出 #2

```
4
```

输入 #3

```
15
```

输出 #3

```
86
```

```
2 -9 -3 2 1 5 -3 4 5 -2 -4 -1 0 3 2
```

说明/提示

【样例解释】

样例1中，最优的 b 为 $[0,1,0,1,2]$，最大答案为 6

样例2中，最优的 b 为 $[0,0,1,0,1]$，最大答案为 4

【数据范围】

对于所有测试点，满足：$1 \le n \le 10^6, -1000 \le a[i] \le 1000$

子任务1（10分）：$n = 5$

子任务2（15分）：$1 \le n \le 10$

子任务3（20分）：$1 \le n \le 500$

子任务4（20分）：$1 \le n \le 5000$

子任务5（5分）：$1 \le n \le 10^6, a[i] \ge 0$

子任务6（30分）：$1 \le n \le 10^6$

本题主要考察搜索、DP、DP优化、和贪心算法。

由于 b[i] 的取值范围是 [0,i-1]，且取小数不优，直接爆搜复杂度是 O(n!)，可以获得25分。

接下来有一个DP，记录：

```
dp[i][j] = b[i]=j时前i个数的最大答案。
```

由于b[i] = j，b[i-1] 的值必须大于等于 j-1，转移：

```
dp[i][j] = max(dp[i-1][j-1...i-2]) + j*a[i]
```

直接实现复杂度O(n^3)，可以获得45分。

不难看出转移是一个后缀max的形式，可以优化为：

```
dp[i][j] = max(dp[i-1][j-1] + j*a[i], dp[i][j+1] - a[i])
```

复杂度 O(n^2)，可以获得65分。另外对于 a[i]>=0 的子任务，直接贪心 b[i] = b[i-1] + 1，这样答案必然最大。

正解是贪心。

由于任何时刻 b[i] 可以回到 0，考虑最优解方案中的 b[i]=0 的位置，这些位置把DP过程分为了互不影响的若干阶段。观察到，最优转移只有两种 b[i]=0（阶段末尾） 或者 b[i]=b[i-1]+1（阶段内部）。

从后往前扫，维护当前阶段的和 sum ，若 sum>0 就 b[i] = b[i-1] + 1，否则就 b[i] = 0 然后 sum = 0（结束当前阶段），最优决策确定后模拟一遍计算答案，复杂度线性。

```
#include<bits/stdc++.h>
#define MAXN 1000005
using namespace std;
typedef long long ll;

int n;
ll a[MAXN], b[MAXN];
int main(){
    cin>>n;
    for(int i=1;i<=n;i++){
        cin>>a[i];
```

```
    }
    ll sum = 0;
    for(int i=n;i>=2;i--){
        sum += a[i];
        if(sum < 0) sum = 0;//结束当前阶段
        else b[i] = 1;
    }
    ll ans = 0;
    for(int i=2;i<=n;i++){
        if(b[i]1){
            b[i] = b[i-1] + 1;
            ans += a[i]*b[i];
        }
    }
    cout<<ans<<endl;
    return 0;
}
```

匹配

https://www.luogu.com.cn/problem/U192222

题目描述

有 n 位毕业生（编号为 $1...n$）正在找工作，同时有 n 家公司（编号为 $1...n$）正在招聘。

每位毕业生心中有一个公司排名，排名越靠前代表他对这个公司越感兴趣；这个排名由二维数组 f 给出，$f[i][j]$ 代表在毕业生 i 心中排名第 j 位的公司编号。

例如 $n=3$:

```
f = [2,3,1
     1,2,3
     2,1,3]
```

则对于 1 号毕业生来说，公司排名为 $f[1]=[2,3,1]$，代表 2 号公司最佳，3 号公司次之，1 号公司最差；对于 2 号毕业生来说，公司排名为 $f[2]=[1,2,3]$；对于 3 号毕业生来说，公司排名为 $f[3]=[2,1,3]$。

同时，每个公司有一个毕业生排名，排名越靠前代表公司认为这个毕业生越优秀。这个排名的由二维数组 g 给出。

最终的目的是让公司和毕业生一一配对（一位毕业生选择恰好一个公司），设当前和 i 号公司配对的毕业生为 $h[i]$，初始 $h[i]$ 都为 0（所有公司全部未配对）。

现在所有毕业生和公司达成协议，他们将重复以下流程，直到配对完成（所有 $h[i]\neq 0$）：

1. 遍历所有毕业生，若 i 号毕业生还没有和任何公司配对，则他会向仍未拒绝过他的公司中最优秀（$f[i]$ 中最靠前的）的公司发出工作申请。
2. 遍历所有公司，若 i 号公司本轮中收到申请，则选出其中最优秀的一个人选（$g[i]$ 中最靠前的），设其为 x。首先公司向除了 x 之外的本轮申请者发出拒绝，然后进行判断：

- 若 $h[i]=0$，则令 $h[i]=x$，即 i 号公司和 x 号毕业生配对；
- 若 $h[i]\neq 0$，且 $h[i]$ 的排名比 x 靠后，则公司取消之前的配对，转而和 x 配对：$h[i]=x$，同时 i 公司向毕业生 $h[i]$ 发出拒绝；
- 若 $h[i]\neq 0$，且 $h[i]$ 的排名比 x 靠前，则 i 公司向毕业生 x 发出拒绝；

3. 此时若所有 $h[i]\neq 0$，则结束；否则回到步骤1。

给定 f,g 两个二维数组，请你编程实现上述过程，输出最终的配对方案 $h[1...n]$。

输入格式

输入一行2个整数，代表 n 和随机种子 $seed$

为了避免输入量过大超时，本题采用以下方式进行输入（你可以复制下面代码作为输入部分）

```
int n,seed;
cin>>n>>seed;

srand(seed);
for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j++) f[i][j] = j;
    random_shuffle(f[i]+1, f[i]+1+n);
}

for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j++) g[i][j] = j;
    random_shuffle(g[i]+1, g[i]+1+n);
}
```

输出格式

输出一行 n 个整数，代表 $h[1...n]$

输入输出样例

输入 #1

```
3 2724
```

输出 #1

```
3 2 1
```

输入 #2

```
10 10
```

输出 #2

```
7 4 6 1 5 8 9 10 2 3
```

输入 #3

```
20 1234
```

输出 #3

```
6 17 1 5 7 20 3 16 18 8 2 15 14 11 12 10 19
```

说明/提示

样例1中 $n = 3$:

```
f = [2,1,3,
     1,2,3,
     2,1,3]

g = [3,1,2,
     2,3,1,
     1,2,3]
```

第一轮开始时，$h = [0, 0, 0]$:

- 1 号毕业生向 2 号公司发出申请
- 2 号毕业生向 1 号公司发出申请
- 3 号毕业生向 2 号公司发出申请
- 1 号公司和 2 号毕业生达成配对
- 2 号公司和 3 号毕业生达成配对，向 1 号毕业生发出拒绝

第二轮开始时，$h = [2, 3, 0]$:

- 1 号毕业生向 1 号公司发出申请
- 1 号公司进行比较，$g[1] = [3, 1, 2]$，1 号比当前 $h[1] = 2$ 排名更靠前，1 号公司和 1 号毕业生达成配对，向 2 号毕业生发出拒绝

第三轮开始时，$h = [1, 3, 0]$:

- 2 号毕业生向 2 号公司发出申请
- 2 号公司进行比较，$g[2] = [2, 3, 1]$，2 号比当前 $h[2] = 3$ 排名更靠前，2 号公司和 2 号毕业生达成配对，向 3 号毕业生发出拒绝

第四轮开始时，$h=[1,2,0]$:

- 3 号毕业生向 1 号公司发出申请
- 1 号公司进行比较，$g[2]=[3,1,2]$，3 号比当前 $h[1]=1$ 排名更靠前，1 号公司和 3 号毕业生达成配对，向 1 号毕业生发出拒绝

第五轮开始时，$h=[3,2,0]$:

- 1 号毕业生向 3 号公司发出申请
- 3 号公司和 1 号毕业生达成配对

第五轮结束后$h=[3,2,1]$，完成一一配对。

【数据范围】

对于所有测试点，$1\le n\le 5000, 0\le seed\le 10^9$，输入方式保证了 f,g 数组中的每行都是 $1,2,\ldots,n$ 的排列

子任务1（20分）：$1\le n\le 10$

子任务2（20分）：$1\le n\le 50$

子任务3（20分）：$1\le n\le 500$

子任务4（20分）：$1\le n\le 1000$

子任务5（20分）：$1\le n\le 5000$

题目就是要你模拟一遍 Gale-Shapley 算法（参见baidu百科）。

本题主要考察对数学文本的阅读能力、以及代码能力。

注意本算法的轮数上界可以达到 O(n^2)，由于数据随机生成，本题没有卡 O(轮数*n) 的做法，并且本题测试点中的轮数大概在 5n-10n 左右，所以只要小心实现就可通过。

以下标程的复杂度是严格 O(n^2)，与轮数无关。

```
#include<bits/stdc++.h>
#define MAXN 5005
using namespace std;
typedef long long ll;
```

```
int n, seed;
int p[MAXN], h[MAXN], vis[MAXN];
int f[MAXN][MAXN], g[MAXN][MAXN];
int rkg[MAXN][MAXN];
vector<int> v1,v2,vt;


void solve(){
 for(int i=1;i<=n;i++) v1.push_back(i);
 while(v1.size()){//v1存放还未匹配的毕业生
   //1
    v2.clear();//v2存放本轮收到申请的公司
   vt.clear();
   for(int k=0;k<v1.size();k++){
     int i = v1[k], j = f[i][++p[i]];
     vis[j] = 0;
   }

   for(int k=0;k<v1.size();k++){
     int i = v1[k], j = f[i][p[i]];
     if(vis[j]==0){
       v2.push_back(j);
       vis[j] = i;
     }
     else if(rkg[j][i] < rkg[j][vis[j]]){
       vt.push_back(vis[j]);
```

```
        vis[j] = i;
      }
      else{
        vt.push_back(i);
      }
    }

    //2
    for(int k=0;k<v2.size();k++){
      int j = v2[k], x = vis[j];
      if(h[j]==0) h[j] = x;
      else if(rkg[j][x] < rkg[j][h[j]]){
        vt.push_back(h[j]);
        h[j] = x;
      }
      else{
        vt.push_back(x);
      }
    }
    swap(v1, vt);
  }
}
int main(){
  srand(time(0));
  cin>>n>>seed;
```

```
  //input
  srand(seed);
  for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j++) f[i][j] = j;
    random_shuffle(f[i]+1, f[i]+1+n);
  }
  for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j++) g[i][j] = j;
    random_shuffle(g[i]+1, g[i]+1+n);
    for(int j=1;j<=n;j++) rkg[i][g[i][j]] = j;
  }
  //solve
  solve();
  //output
  for(int i=1;i<=n;i++) cout<<h[i]<<" ";
  return 0;
}
```