A codeforces-731B:

题意：买披萨，有两种优惠方式，一天买两块，连续两天买一块，问是否每一块披萨都优惠

思路： 两种优惠方式其实都是买两块，区别是天数，读入数据，当遇见0的时候，判断之前买的披萨总和是否整除2，是则都优惠到了

代码：

#include <map>

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

int main(){

int n,a,sum=0,flag=1;

scanf("%d",&n);

for(int i=0;i<n;i++){

scanf("%d",&a);

if(a==0){

if(sum%2){

flag=0;

break;

}

sum=0;

}

sum+=a;

}

if(sum%2) flag=0;

flag?printf("YES\n"):printf("NO\n");

}

B codeforces-731C

题意：有n支不同颜色的袜子，穿m天，袜子的颜色数为k，每天穿的袜子的序号固定，问最少染色几支袜子使得每天穿的袜子的颜色一样

思路：因为之前穿过的袜子可能继续穿，每个数据相互制约，所以使用并查集合并每天穿的袜子，记录颜色，将每个集合染成出现次数最多的那个颜色，即可保证每次染最少的袜子

代码：

#include <map>

#include <vector>

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

int col[200010],fa[200010]; //记录颜色，保存并查集关系树

vector<int> sock[200010]; //设置集合，并保存颜色

int find(int x){

if(fa[x] == x) return fa[x];

return fa[x]=find(fa[x]);

}

void unit(int a,int b){

a=find(a);

b=find(b);

if(a!=b) fa[a]=b;

}

int main(){

int n,m,k;

scanf("%d%d%d",&n,&m,&k);

for(int i=1;i<=n;i++){

scanf("%d",&col[i]);

fa[i] = i;

}

int a,b;

for(int i=1;i<=m;i++){

scanf("%d%d",&a,&b);

unit(a,b);

}

for(int i=1;i<=n;i++)

sock[find(i)].push\_back(col[i]); //记录每个集合元素的颜色

int ans=0;

for(int i=1;i<=n;i++){

if(sock[i].size()<=1) continue;

int maxn=-1;

map<int,int> cnt;

cnt.clear();

for(int j=0;j<sock[i].size();j++){

cnt[sock[i][j]]++;

maxn=max(maxn,cnt[sock[i][j]]); //选取出现次数最多的颜色

}

ans+=sock[i].size()-maxn;

}

printf("%d\n",ans);

return 0;

}

这题也应该可以用map来写，但是我在26组数据时爆了内存，后来改用vector但是发现TLE错误，原来是find函数写错了…..引以为戒

C codeforces-731F

题意：在一串数中找一个leading，使得大于leading的树减小成为leading的倍数，并且和最大

思路：暴力，每一个数都假设成为leading，介于i\*j -> i\*(j+1)-1之间的数字都需要变成i\*j，但是如果每一个数都遍历一遍是否为leading的倍数，肯定会T，所以可以每次求一个区间的和，通过统计出现的次数，将一个区间的次数保存下来，即可每次求一个区间的数的个数

代码：

#include <map>

#include <vector>

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

long long cnt[400005],sum\_cnt[400005];

int main(){

int n;

scanf("%d",&n);

for(int i=0;i<n;i++){

long long tmp;

scanf("%lld",&tmp);

cnt[tmp]++;

}

long long ans=0;

for(int i=200000;i>0;i--)

sum\_cnt[i]=sum\_cnt[i+1]+cnt[i]; //统计比i小的数的次数

for(int i=1;i<200001;i++)

if(cnt[i]>0){

long long sum=0;

for(int j=i;j<200001;j+=i) //区间移动

sum+=(sum\_cnt[j]-sum\_cnt[j+i])\*j; //区间求和

ans=max(sum,ans);

}

printf("%lld",ans);

}

D codeforces-919B

题意：求最小完美数，完美数即各个位数和为10

思路：从1开始判断，取出每一位相加判断

代码：

#include <map>

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

int cal(int num){

    int ans=0;

    while (num){

        ans+=num%10;

        num/=10;

    }

    return ans;

}

int main(){

int n;

    scanf("%d",&n);

    int ans=0;

    while(n){

        ans++;

        if(cal(ans)==10) n--;

    }

    printf("%d\n", ans);

}

E codeforces-919C

题意：n\*m的矩阵，找k个空座位处于同一列或同一行，问有几种情况

思路：暴力，行列分别遍历，当k=1时，同行同列情况相同，只计算一种

代码：

#include <vector>

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

string s[2005];

int main(){

int n,m,k;

scanf("%d%d%d",&n,&m,&k);

for(int i=0;i<n;i++) cin>>s[i];

int sum=0;

for(int i=0;i<n;i++){

int t=0;

for(int j=0;j<m;j++){

if(s[i][j]=='.') t++;

else{

if(t>=k) sum+=t-k+1;

t=0;

}

}

if(t>=k) sum+=t-k+1;

}

if(k!=1){

for(int i=0;i<m;i++){

int t=0;

for(int j=0;j<n;j++){

if(s[j][i]=='.') t++;

else{

if(t>=k) sum+=t-k+1;

t=0;

}

}

if(t>=k) sum+=t-k+1;

}

}

printf("%d\n",sum);

return 0;

}

F codeforces 884B

题意：猜题意，我的理解是给出x个位置，有n个区间，每一个区间大小给出，有唯一的区间字段即每个区间由一个0隔开

思路：减去每个区间的占位数，若得到n-1个空位，既符合题意

代码：

#include <map>

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

int main(){

int n,x;

scanf("%d%d",&n,&x);

for(int i=0;i<n;i++){

int tmp;

scanf("%d",&tmp);

x-=tmp;

}

x==n-1?printf("YES\n"):printf("NO\n");

}

G codeforces-884C

题意：给出n个数，第i个站可以到达pi站，题目保证所有的点都可以由其中一个点到达，例如样例2：

5

1 5 4 3 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | 5 | 4 | 3 | 2 |

由上表可得 1连接1；2，5相互连接；3，4相互连接

题目规定每两个站台连接，方便数加1，现在可改变两个站台的连接方向，求最大方便数

思路：因为题目保证每个站台都处于一个环中，而每个环两个点对的个数为n\*n(n为点的个数），而改变两个点的连接方向可以合并两个环，根据重要不等式：记两数分别是a,b，则有(a+b)^2>=a^2+b^2,则将两个最大的环联通可得最大点对数。

代码：

#include <vector>

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

vector<int> c;

int a[100010];

bool vis[100010];

int main(){

c.clear();

memset(vis,0,sizeof(vis));

long long n;

cin>>n;

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

for(int i=1;i<=n;i++){

if(vis[i]==1) continue;

int cnt=1;

vis[i]=1;

int tmp=a[i];

while(tmp!=i){ //判断环的大小并记录

vis[tmp]=1;

tmp=a[tmp];

cnt++;

}

c.push\_back(cnt);

}

long long ans=0;

sort(c.begin(),c.end());

if(c.size()>=2){

ans=c[c.size()-1]+c[c.size()-2]; //合并两个最大环

ans\*=ans;

for(int i=0;i<c.size()-2;i++) ans+=c[i]\*c[i];//计算其余环

}

else ans=n\*n;

cout<<ans<<endl;

}

H codeforces-884A

题意：n天，读书花费t的时间，每天工作Ai的时间，求读完书的天数

思路：总读书的时间减去每天剩余的时间，t<0时读完

代码：

#include <map>

#include <stdio.h>

#include <string.h>

#include <iostream>

#include <algorithm>

using namespace std;

int main(){

int n,m;

scanf("%d%d",&n,&m);

int sum=0,tmp;

for(int i=0;i<n;i++){

scanf("%d",&tmp);

m-=(86400-tmp);

if(m<=0){

printf("%d",i+1);

break;

}

}

}