Div2的C,D题目（模板开始），可以倒开

Div3第6，7题目

天梯赛5.6月

年底牛客寒假训练营（自己付费）

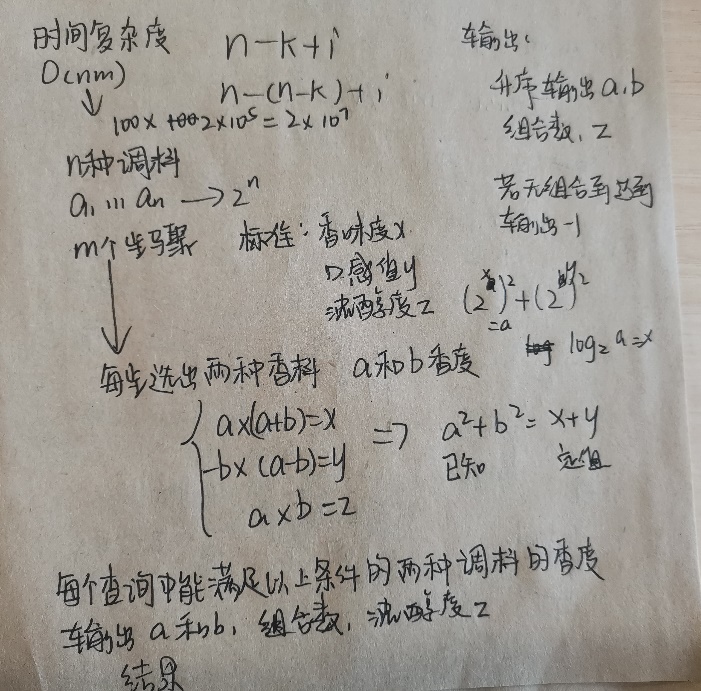
广东省省赛5.6道题目要突破

一眼看出这题目用什么方法

### 原题1

[G-灵魂之汁，浇给~\_长沙学院2024年程序设计竞赛春季校赛 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84832/G)

思路：



1首先.香度的范围是1≤*ai*​≤109) ，用at[]数组记录每个香度的出现次数，但会造成内存的溢出，但是根据题目中每个香度符合2的n次方，根据这个特性，用f()函数计算出指数n,不会造成内存的溢出，时间复杂度也大大减少

2.*a*×(*a*+*b*)=*x*,−*b*×(*a*−*b*)=*y* 解方程得出

3.不用指数函数pow(n,m)而是自己写一个pow（）原因可能是long long

4.组合数根据题目给的样例2得出规律来猜测

代码：

1. #include<cstdio>
2. #include<cstring>
3. #include<bits/stdc++.h>
4. using namespace std;
5. #define int long long
6. const int N=2e5+10;
7. int a[N],at[35],x[110],y[110];
8. int pow(int x){
9. int s=1;
10. while(x--){
11. s\*=2;
12. }
13. return s;
14. }
15. signed f(int x)
16. {
17. int s=0;
18. while(x>1)
19. {
20. x/=2;
21. s++;
22. }
23. return s;
24. }
25. signed main()
26. {
27. int n,m,num;
28. cin>>n>>m;
29. memset(at,0,sizeof(at));
30. for(int i=1;i<=n;i++)
31. {
32. cin>>a[i];
33. at[f(a[i])]++;
34. }
35. for(int g=1;g<=m;g++)
36. {
37. cin>>x[g]>>y[g];
38. int flag=0,i,j;
39. for(i=0;i<35;i++)
40. {
41. for(j=0;j<35;j++)
42. {
43. if(at[i]!=0 && at[j] != 0)
44. {
45. if(pow(2\*i)+pow(2\*j) == x[g]+y[g])
46. {
47. flag=1;
48. break;
49. }
50. }
51. }
52. if(flag==1)
53. break;
54. }
55. if(flag==0)
56. {
57. printf("-1\n");
58. }
59. else
60. {
61. if(i==j)
62. {
63. num=at[i]\*(at[j]-1)/2;
64. }
65. else
66. {
67. num=at[i]\*at[j];
68. }
69. printf("%lld %lld %lld %lld\n",pow(i),pow(j),num,pow(i)\*pow(j));
70. }
71. }
72. }

### 原题2

[B-困兽之斗\_长沙学院2024年程序设计竞赛春季校赛 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84832/B)

思路：

1. 首先要明白中位数是什么：将一组数据按照由小到大（或由大到小）的顺序排列，如果数据的个数是奇数，则称处于中间位置的数为这组数据的中位数；如果数据的个数是偶数，则称中间两个数据的平均数为这组数据的中位数.
2. 根据题意“一段时间内缩小体型的法术，每次可以减少自己的身高 **或** 体重 1 单位量，最小减至 1 ”，写个双循环从1逐渐加到相应的身高h或体重w，判断h/w符不符合比例在r~k,符合再计算出相应的魔法体积(h∗1.9+w)/(0.3∗w)，存入数组v中
3. 注意细节！！！i/j中i和j都是整数，计算出来也是整数，所以要强制转换(double)。然后浮点数不能比较是否相等，所以i/j多乘100

代码：

1. #include<cstdio>
2. #include<bits/stdc++.h>
3. using namespace std;
4. int main()
5. {
6. int h,w,k,r;
7. cin>>h>>w>>k>>r;
8. double v[200000];
9. int cnt=0;
10. for(int i=1;i<=h;i++)
11. {
12. for(int j=1;j<=w;j++)
13. {
14. double a=(double)i/j;
15. double b = a \* 100;
16. if(r<=b && b<=k)
17. {
18. double sum=(i\*1.9+j)/(0.3\*j);
19. v[cnt]=sum;
20. cnt++;
21. }
22. }
23. }
24. sort(v,v+cnt);
25. if(cnt%2==1)
26. {
27. printf("%.2lf",v[cnt/2]);
28. }
29. else if(cnt%2==0)
30. {
31. printf("%.2lf",(v[cnt/2]+v[cnt/2-1])/2.0);
32. }

35. }

### 原题3

[B-茉茉的密码\_牛客周赛 Round 47 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84851/B)

思路：

脑筋急转弯，找n个字符串的任意一个公共子串，一个字母也可以当子串，那么便从26个字母找，两个标记数组，一个标记数组用来记录26个字母在每个字符串是否出现，另一个标记数组防止在同一个字符串中加多了

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. int ch1[50],ch2[50];
4. int main()
5. {
6. int n;
7. cin>>n;
8. memset(ch1,0,sizeof(ch1));
9. for(int i=0;i<n;i++)
10. {
11. string s;
12. cin>>s;
13. memset(ch2,0,sizeof(ch2));
14. for(int j=0;j<s.size();j++)
15. {
16. int c=s[j]-'a';
17. if(ch2[c]==0)
18. {
19. ch1[c]++;
20. ch2[c]++;
21. }
23. }
24. }
25. for(int i=0;i<50;i++)
26. {
27. if(ch1[i]==n)
28. {
29. printf("%c",i+'a');
30. break;
31. }
32. }

35. }

### 原题4

[C-苗苗的气球\_牛客周赛 Round 47 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84851/C)

思路：

气球最后只能剩下一种或没有，那么考虑：只有两种一样数量的气球，不管奇数还是偶数，最后会同归于尽，即0种；最大数的气球比其它气球的总和还大的话，最后只能剩下最大数气球，即1种；总和为奇数，总能剩下一种气球，每种气球都有可能剩下，即n种；总数为偶数，最后要剩下来的那种气球的个数必须为偶数，这样剩下其他种类的偶数会同归于尽，那么要达到留下来的条件是偶数不能是1，必须比1大，所有记录有多少个1，那么剩下来气球种数可能是为非1种的个数，即n-cnt

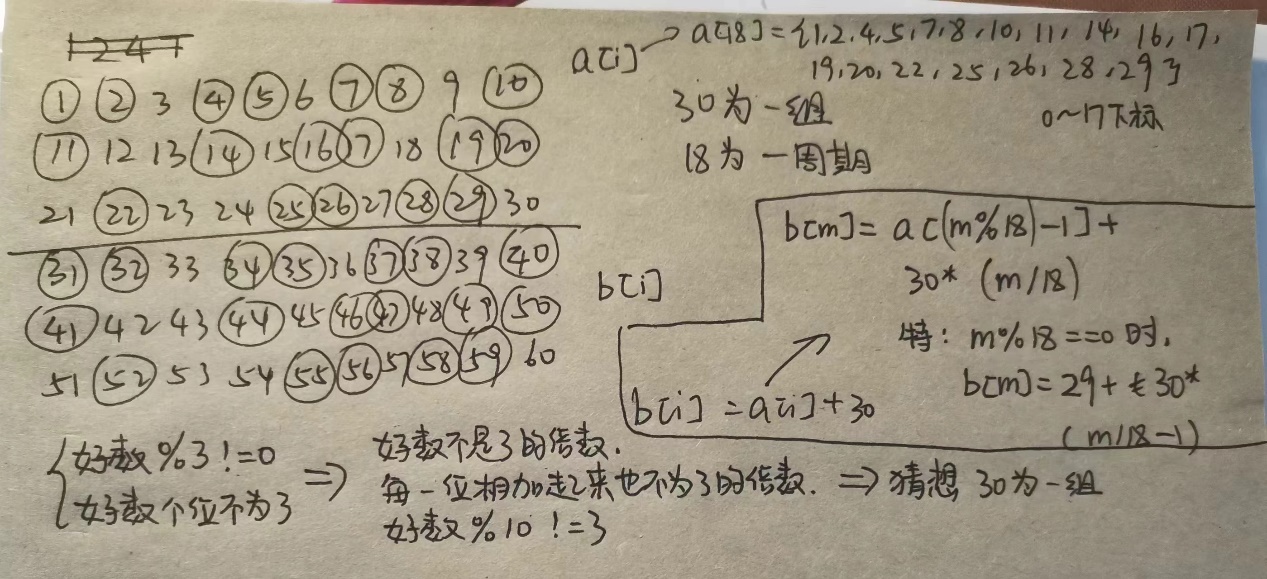
代码：

1. #include<cstdio>
2. #include<bits/stdc++.h>
3. using namespace std;
4. #define int long long
5. signed main()
6. {
7. int n;
8. cin>>n;
9. int a[n];
10. int sum=0,cnt=0;
11. for(int i=0;i<n;i++)
12. {
13. cin>>a[i];
14. sum+=a[i];
15. if(a[i]==1)
16. cnt++;
17. }
18. sort(a,a+n);
19. int maxn=a[n-1];
20. if(n==2 && a[0]==a[1])
21. {
22. printf("0");
23. }
24. else if(maxn>sum-maxn)
25. {
26. printf("1");
27. }
28. else if(sum%2==1)
29. {
30. printf("%d",n);
31. }else{
32. printf("%d",n-cnt);
33. }
34. }

### 原题5

[D-萌萌的好数\_牛客周赛 Round 47 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84851/D)

思路：



30划定范围，18为一周期，b[i]=a[相应的下标]+30\*周期

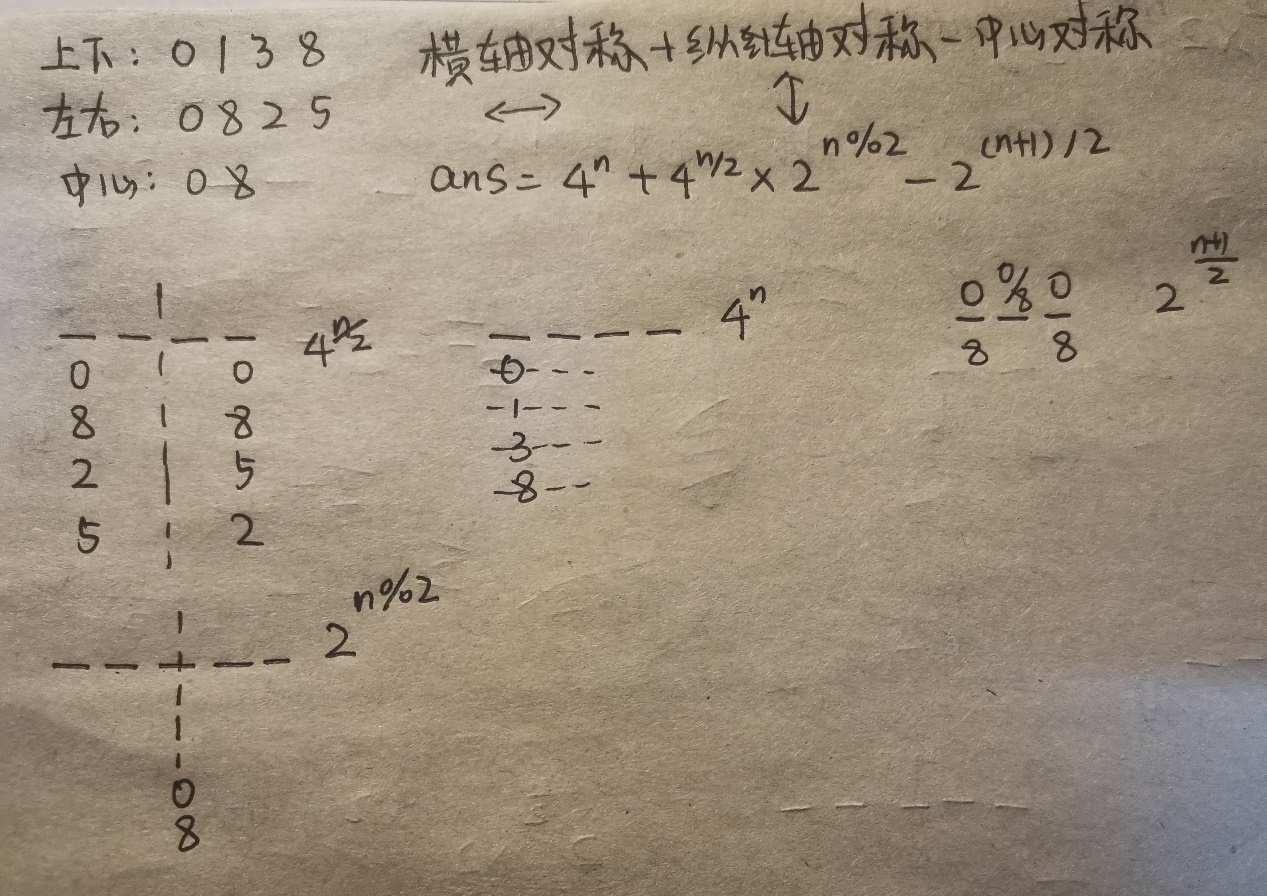
代码：

1. #include<cstdio>
2. #include<bits/stdc++.h>
3. using namespace std;
4. #define int long long
5. int a[18]={1, 2, 4, 5, 7, 8, 10, 11, 14, 16, 17, 19, 20, 22, 25, 26, 28, 29};
6. signed main()
7. {
8. int t;
9. cin>>t;
10. while(t--)
11. {
12. int n;
13. cin>>n;
14. int ans;
15. if(n%18==0)
16. ans=29+30\*(n/18-1);
17. else
18. ans=a[(n%18)-1]+30\*(n/18);
19. cout << ans << endl;
20. }
21. }

### 原题6

[E-茜茜的计算器\_牛客周赛 Round 47 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84851/E)

思路：



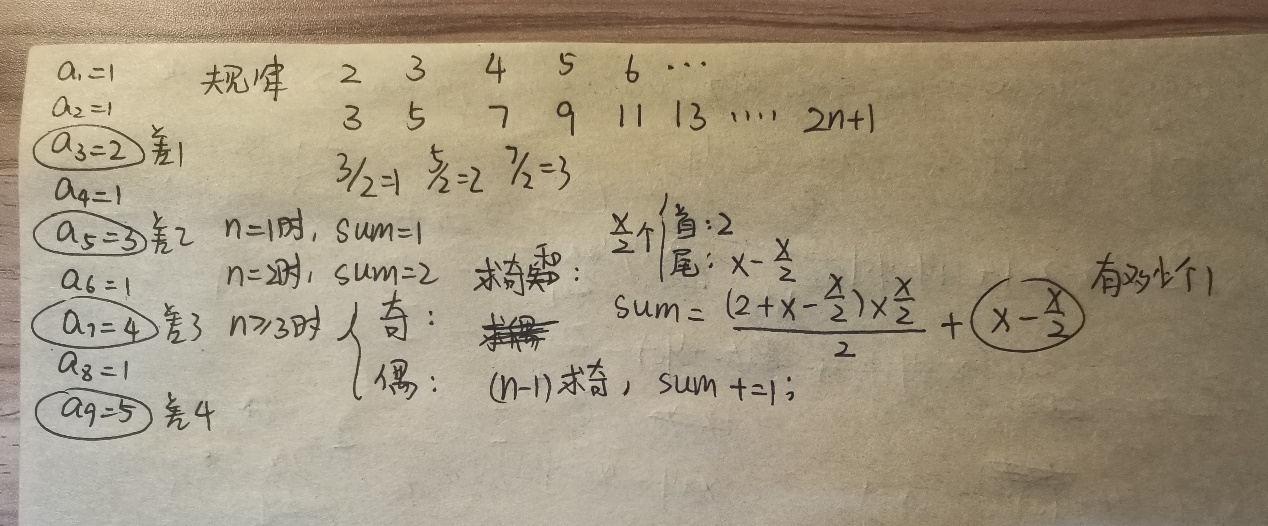
代码：

1. #include<cstdio>
2. #include<bits/stdc++.h>
3. using namespace std;
4. #define int long long
5. const int N=1e9+7;
6. int qpow(int x,int y)
7. {
8. int sum=1;
9. while(y)
10. {
11. if(y&1)
12. sum = sum \* x % N;
13. x = x \* x % N;
14. y >>= 1;
15. }
16. return sum;
17. }
18. signed main()
19. {
20. int n;
21. cin>>n;
22. int ans=( ( qpow(4,n) + qpow(4,n/2) \* qpow(2,n%2) - qpow(2,(n+1)/2) ) % N + N ) % N;
23. printf("%lld",ans);
25. }

### 原题7

1. [魔法序列\_牛客挑战赛75 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84888/A)

思路：



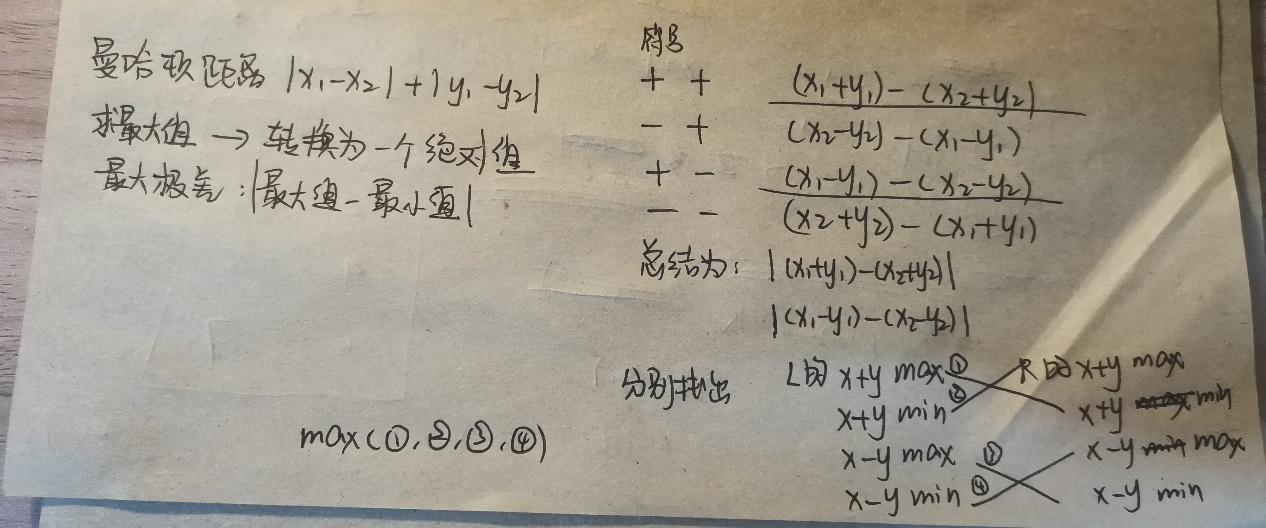
代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. int solve(int x)
5. {
6. int num=x/2;
7. int sum=((2+x-num)\*num)/2;
8. return sum+x-num;
9. }
10. signed main()
11. {
12. int n;
13. scanf("%lld",&n);
14. if(n==1)
15. printf("1");
16. else if(n==2)
17. printf("2");
18. else{
19. if(n&1){
20. printf("%lld",solve(n));
21. }
22. else{
23. int x=n-1;
24. int sum=1+solve(x);
25. printf("%lld",sum);
26. }
28. }
29. }

### 原题8

[B-赛跑\_牛客挑战赛75 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84888/B)

思路：



代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. const int N=1e5+10;
5. int sum1[N],sum2[N],sum3[N],sum4[N];
6. signed main()
7. {
8. int n,m;
9. scanf("%lld%lld",&n,&m);
10. for(int i=0;i<n;i++){
11. int x,y;
12. scanf("%lld%lld",&x,&y);
13. sum1[i]=x+y;
14. sum3[i]=x-y;
15. }
16. for(int i=0;i<m;i++){
17. int x,y;
18. scanf("%lld%lld",&x,&y);
19. sum2[i]=x+y;
20. sum4[i]=x-y;
21. }
22. sort(sum1,sum1+n);
23. sort(sum2,sum2+m);
24. sort(sum3,sum3+n);
25. sort(sum4,sum4+m);
26. int ans1=abs(sum1[n-1]-sum2[0]);
27. int ans2=abs(sum2[m-1]-sum1[0]);
28. int ans3=abs(sum3[n-1]-sum4[0]);
29. int ans4=abs(sum4[m-1]-sum3[0]);
30. int ans=max( ans4, max( ans3,max( ans1,ans2) ) );
31. printf("%lld",ans);
32. }

### 原题9

[Problem - D - Codeforces](https://codeforces.com/contest/1992/problem/D)

思路：模拟

在给定的字符串中的每个字符有三种情况：L、W、C，可以跳m段，游k段

用**计数器**jump,swim记录**目前**可跳次数和可游次数jump=m-1,swim=k，如果能将字符串完全循环一遍，那就是YES

若为L：计数器jump重置为m-1

若为W：若jump次数够，选跳jump--，否则若swim次数够，选游swim--;否则，退出循环

若为C：若jump次数够，选跳jump--;否则，退出循环

若循环能完全走完，就输出“YES”，不能就输出“NO”

代码1：

1. #include<bits/stdc++.h>
2. using namespace std;
3. int main()
4. {
5. int t;
6. cin>>t;
7. while(t--)
8. {
9. int n,m,k;
10. cin>>n>>m>>k;
11. string ch;
12. cin>>ch;
13. int jump=m-1,swim=k,i;
14. for(i=0;i<n;i++)
15. {
16. if(ch[i]=='C')
17. {
18. if(jump>0)
19. jump--;
20. else
21. break;
22. }
23. else if(ch[i]=='W')
24. {
25. if(jump>0)
26. jump--;
27. else if(swim>0)
28. swim--;
29. else
30. break;
31. }
32. else if(ch[i]=='L')
33. jump=m-1;
34. }
35. if(i==n)
36. cout<<"YES"<<endl;
37. else
38. cout<<"NO"<<endl;
39. }
40. }

还有另一个方法，就是找出此岸的坐标和原木的坐标和对岸的坐标，记录下来，之间的间距要是小于等于m（即可以最大跳的段数）就可以从上一个原木/岸跳到下一个原木/岸，要是间距超过m，那就从上一个原木/岸往前跳m段，再逐个字符到下一个原木/岸，这个过程要是有C就“NO”，要是有W，若能游泳的段数不超过k，那就游过去，否则“NO”

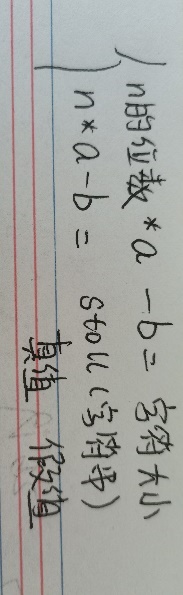
代码2：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define long long int
4. signed main()
5. {
6. int t;
7. cin>>t;
8. while(t--)
9. {
10. int n,m,k;
11. cin>>n>>m>>k;
12. string s;
13. cin>>s;
14. s=" "+s;
15. vector<int> a;
16. int flag=1;
17. a.push\_back(0);
18. for(int i=0;i<s.size();i++)
19. {
20. if(s[i]=='L')
21. a.push\_back(i);
22. }
23. a.push\_back(n+1);
24. for(int i=1;i<a.size();i++)
25. {
26. if(a[i]-a[i-1]<=m)
27. continue;
28. else
29. {
30. for(int j=a[i-1]+m;j<a[i];j++)
31. {
32. if(s[j]=='C')
33. flag=0;
34. else if(s[j]=='W')
35. {
36. if(k>0)
37. k--;
38. else
39. flag=0;
40. }
41. }
42. }
43. }
44. if(!flag)
45. cout<<"NO"<<endl;
46. else
47. cout<<"YES"<<endl;
48. }
49. }

### 原题10

[Problem - E - Codeforces](https://codeforces.com/contest/1992/problem/E)

思路：枚举



其中

100\*10000=100 0000，所以真值最大为6位数，即字符个数最大为6，从1枚举到6

a从1枚举到10000，b根据如图第一个式子得出

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define long long int
4. signed main()
5. {
6. int t;
7. cin>>t;
8. while(t--)
9. {
10. int n;
11. cin>>n;
12. vector< pair<int,int> > g;
13. string x=to\_string(n);
14. int wei=x.size();
15. while(x.size()<6)
16. x+=x;
17. for(int a=1;a<=10000;a++)*//枚举*
18. {
19. for(int b=wei\*a-1;1<=b && b<=min(10000,a\*n) && b>=wei\*a-6; b--)
20. {
21. int res=wei\*a-b;
22. string s=x.substr(0,res);
23. if(n\*a-b==stoll(s))
24. g.push\_back({a,b});
25. }
26. }
27. cout<<g.size()<<endl;
28. for(auto l:g)
29. {
30. cout<<l.first<<" "<<l.second<<endl;
31. }
32. }
33. }

### 原题11

[100368. 从链表中移除在数组中存在的节点 - 力扣（LeetCode）](https://leetcode.cn/problems/delete-nodes-from-linked-list-present-in-array/)

思路：模板，单链表的删除

代码：

1. */\*\**
2. \* Definition for singly-linked list.
3. \* struct ListNode {
4. \*     int val;
5. \*     ListNode \*next;
6. \*     ListNode() : val(0), next(nullptr) {}
7. \*     ListNode(int x) : val(x), next(nullptr) {}
8. \*     ListNode(int x, ListNode \*next) : val(x), next(next) {}
9. \* };
10. \*/
11. class Solution {
12. public:
13. ListNode\* modifiedList(vector<int>& nums, ListNode\* head) {
14. set<int> num( nums.begin(),nums.end() );
15. //把要求删除数数组装进集合里，到时候检查某个数是否存在于集合里，是的话删了
16. ListNode\* list=new ListNode(0,head);//创建0头结点，并指向head链表
17. ListNode\* current=list//现在的指针;
18. while(current->next != NULL){
19. if(num.count(current->next->val)){//删了
20. ListNode\* temp=current->next;
21. current->next=temp->next;
22. delete temp;
23. }else{
24. current=current->next;
25. }
26. }
27. ListNode\* ans=list->next;
28. delete list;
29. return ans;
30. }
31. };

### 原题12

[100361. 切蛋糕的最小总开销 I - 力扣（LeetCode）](https://leetcode.cn/problems/minimum-cost-for-cutting-cake-i/description/)

思路：贪心，横着切蛋糕会影响竖着切蛋糕，竖着切蛋糕会影响横着切蛋糕

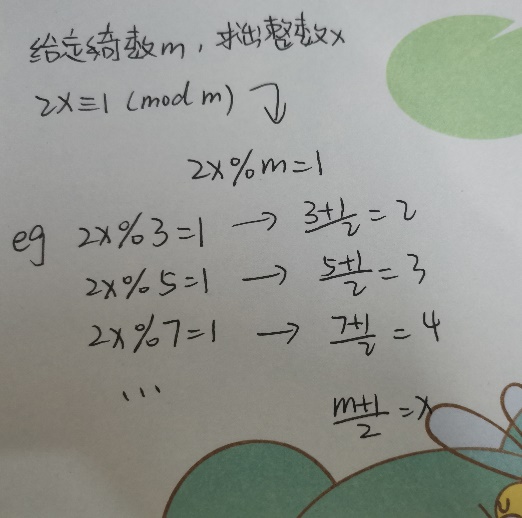
代码：

1. class Solution {
2. public:
3. int minimumCost(int m, int n, vector<int>& horizontalCut, vector<int>& verticalCut) {
4. sort(horizontalCut.begin(),horizontalCut.end(),greater<int>());
5. sort(verticalCut.begin(),verticalCut.end(),greater<int>());
6. int h=0,v=0;
7. int hcut=1,vcut=1;
8. int sum=0;
9. //双指针，两端区间，谁先没了循环结束
10. while(h<horizontalCut.size() && v<verticalCut.size()){
11. if(horizontalCut[h]>=verticalCut[v]){
12. sum+=horizontalCut[h]\*hcut;
13. h++;
14. vcut++;
15. }
16. else{
17. sum+=verticalCut[v]\*vcut;
18. v++;
19. hcut++;
20. }
21. }
22. *//扫尾工作*
23. while(h<horizontalCut.size()){
24. sum+=horizontalCut[h]\*hcut;
25. h++;
26. vcut++;
27. }
28. while(v<verticalCut.size()){
29. sum+=verticalCut[v]\*vcut;
30. v++;
31. hcut++;
32. }
33. return sum;
34. }
35. };

### 原题13

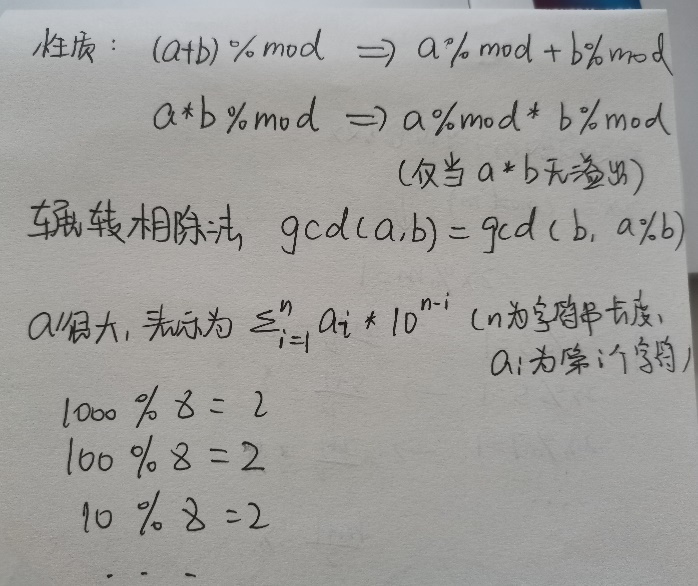
1. [小红的同余\_牛客周赛 Round 51 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/86034/A)

思路：找规律



### 原题14

[D-小红的 gcd\_牛客周赛 Round 51 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/86034/D)

思路： 

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. signed main()
5. {
6. string a;
7. int b;
8. cin>>a>>b;
9. int sum=0;
10. for(int i=0;i<a.size();i++){
11. sum =sum\*10 +a[i]-'0';//a的每一位可%b
12. sum =sum%b; //辗转相除法gcd(a,b)=gcd(b,a%b)
13. }
14. cout<<\_\_gcd(b,sum);
15. }

### 原题15

[Problem - A - Codeforces](https://codeforces.com/contest/1988/problem/A)

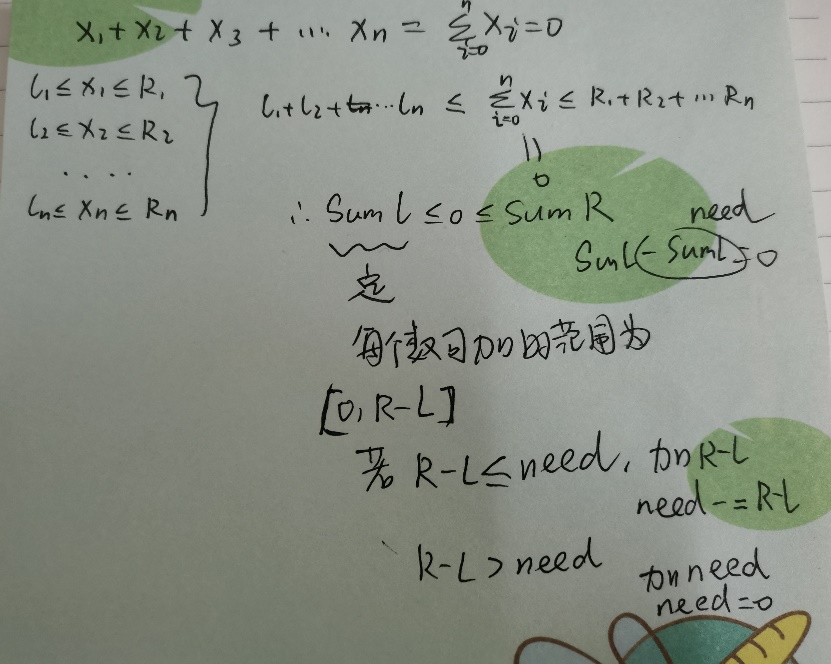
思路：拆出来k-1个1，最后一位继续拆，直到全为1

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. int main()
4. {
5. int t;
6. cin>>t;
7. while(t--)
8. {
9. int n,k;
10. cin>>n>>k;
11. int ans=0;
12. while(n>1){
13. n-=k-1;//全拆为1，剩下一位填上去，直到大家都是1
14. ans++;
15. }
16. cout<<ans<<endl;
17. }
18. }

### 原题16

[C - Sum = 0 (atcoder.jp)](https://atcoder.jp/contests/abc362/tasks/abc362_c)

思路：

将每个答案先定为最小数L，再增加直到和为0

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. signed main()
5. {
6. int n;
7. cin>>n;
8. int l[n],r[n];
9. int suml=0,sumr=0;
10. for(int i=0;i<n;i++)
11. {
12. cin>>l[i]>>r[i];
13. suml+=l[i],sumr+=r[i];
14. }
15. if(suml<=0 && 0<=sumr)//符合范围，可以找到答案
16. {
17. cout<<"Yes"<<endl;
18. int need=-suml;//努力找到持平sum1
19. for(int i=0;i<n;i++)
20. {
21. int ans=l[i];//每次答案先定为左范围
22. int temp=r[i]-l[i];
23. if(temp<=need)//能持平一点是一点
24. {
25. ans+=temp;
26. need-=temp;
27. }
28. else{//持平了，不管了
29. ans+=need;
30. need=0;
31. }
32. cout<<ans<<endl;
33. }
34. }
35. else{
36. cout<<"No"<<endl;
37. }
38. }

### 原题17

[Problem - B - Codeforces](https://codeforces.com/contest/1994/problem/B)

思路：模拟，要是第一个字符串1之前为0，第二个字符串需要1就“NO”了

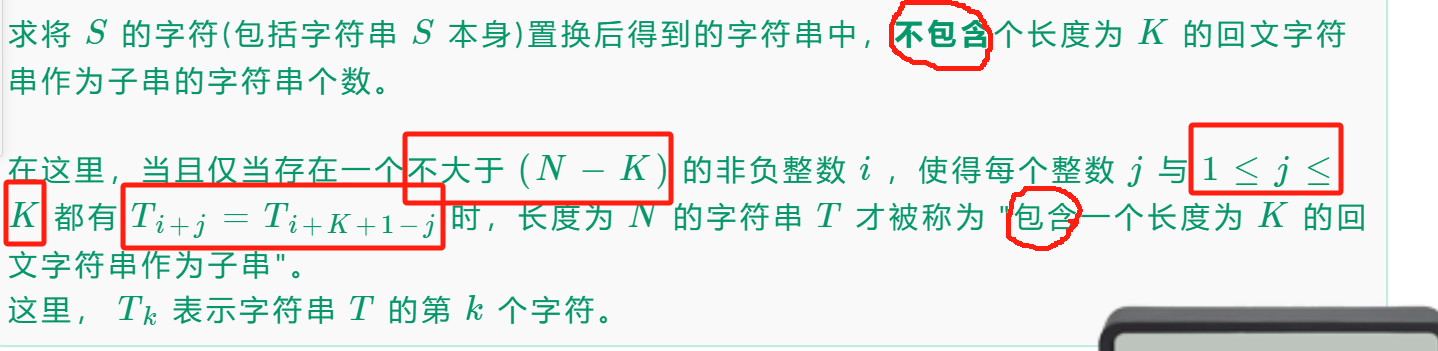
代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. signed main()
5. {
6. int t;
7. cin>>t;
8. while(t--)
9. {
10. int n;
11. cin>>n;
12. string s,t;
13. cin>>s>>t;
14. if(s.find('1')>t.find('1'))//第一字符串首次出现的1比第二个字符串首次出现的1还后，代表不行
15. cout<<"NO"<<endl;
16. else
17. cout<<"YES"<<endl;
18. }
19. }

### 原题18

[C - Avoid K Palindrome 2 (atcoder.jp)](https://atcoder.jp/contests/abc363/tasks/abc363_c)

思路：



对字符串进行全排列（对升序数组），找出“包含长度为k的回文字符子串”

0<=i<=n-k（i是定下来的），0<=j<=k，双指针

Ti+j=Ti+k-1-j，对称的

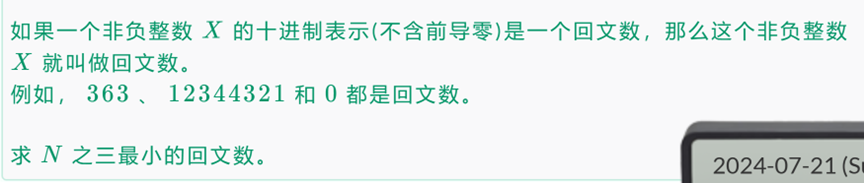
代码：

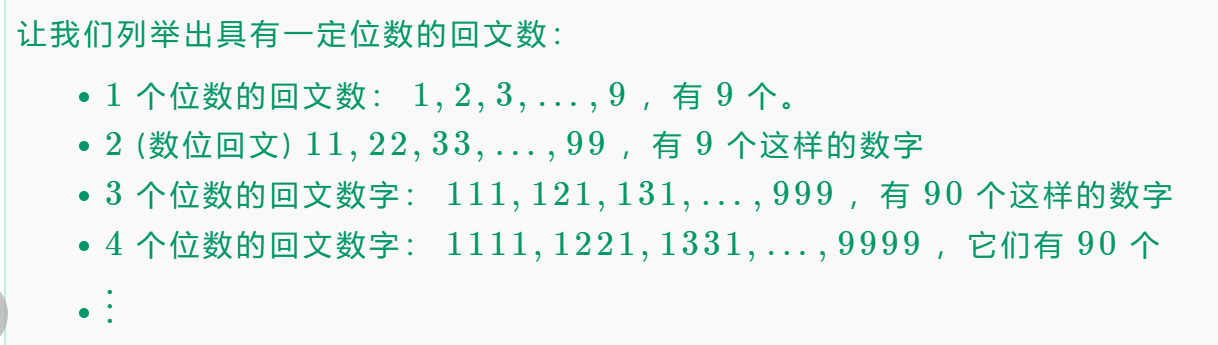
1. #include<bits/stdc++.h>
2. using namespace std;
3. int main()
4. {
5. int n,k;
6. cin>>n>>k;
7. string s;
8. cin>>s;
9. sort(s.begin(),s.end());//对升序数组才能进行全排列
10. int ans=0;
11. do{
12. int flag=1;
13. for(int i=0;i<=n-k;i++)
14. {
15. int cnt=0;
16. for(int j=0;j<k;j++)
17. {
18. if(s[i+j]==s[i+k-1-j])
19. cnt++;
20. }
21. if(cnt==k)//找到含有长度为k的回文字符串子串
22. flag=0;
23. }
24. if(flag==1)//没找到，符合要求
25. ans++;
26. }while(next\_permutation(s.begin(),s.end()));
27. cout<<ans<<endl;
28. }

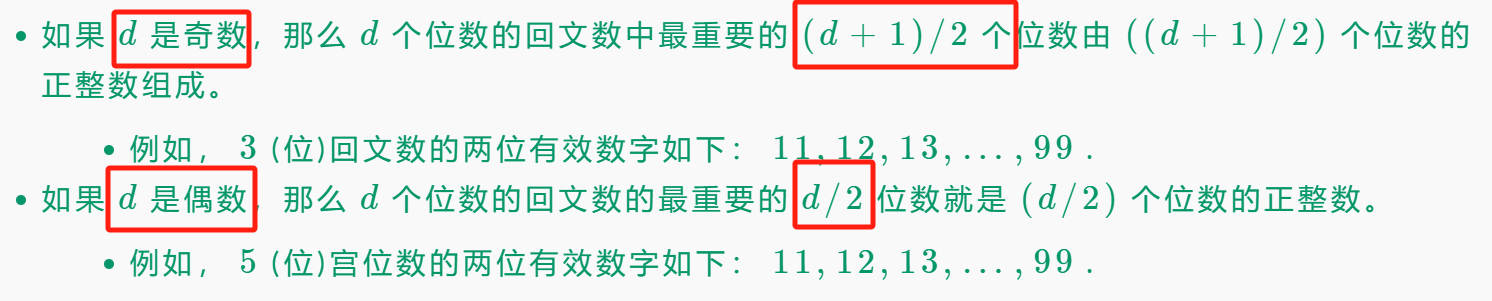
### 原题19

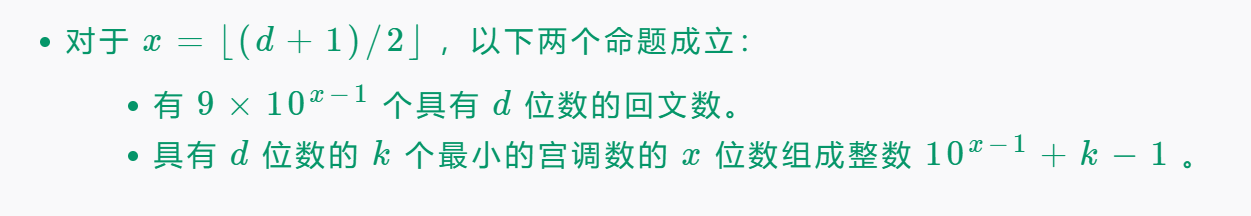
[D - Palindromic Number (atcoder.jp)](https://atcoder.jp/contests/abc363/tasks/abc363_d)

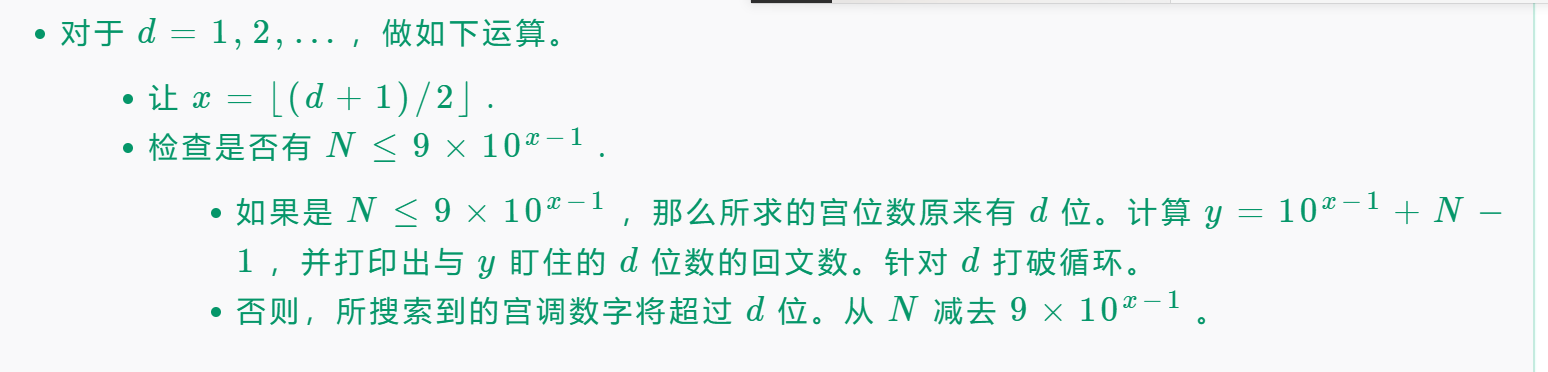
思路：求第k个回文数，1≤N≤1e18











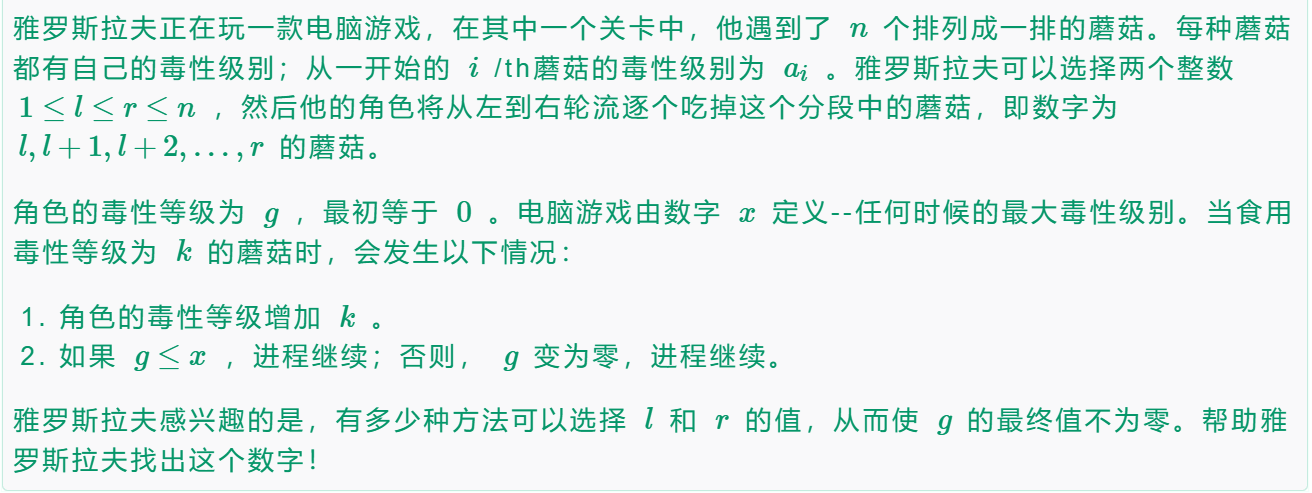
复杂度为 O(logN) ，所求的回文数可能有多达 35 位数，应使用字符串类型

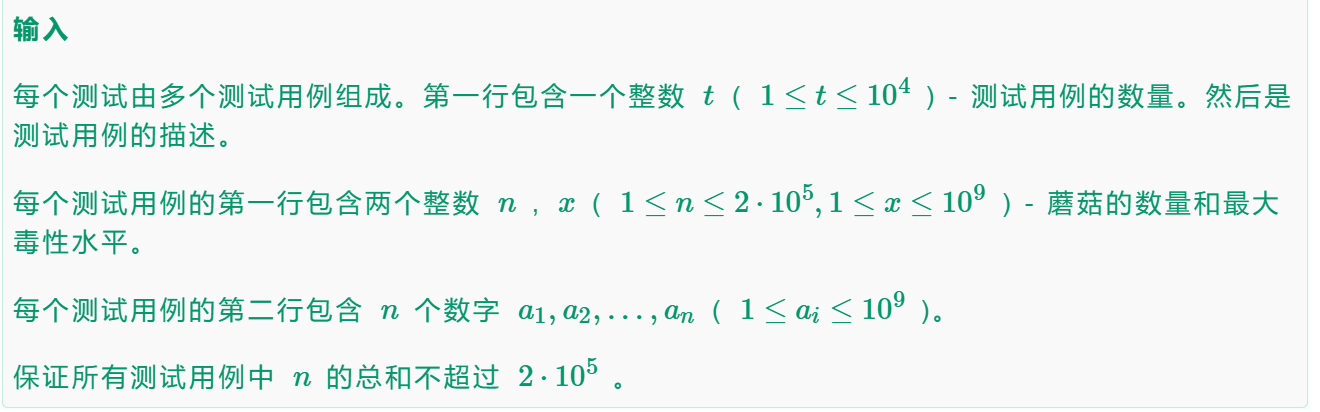
代码：

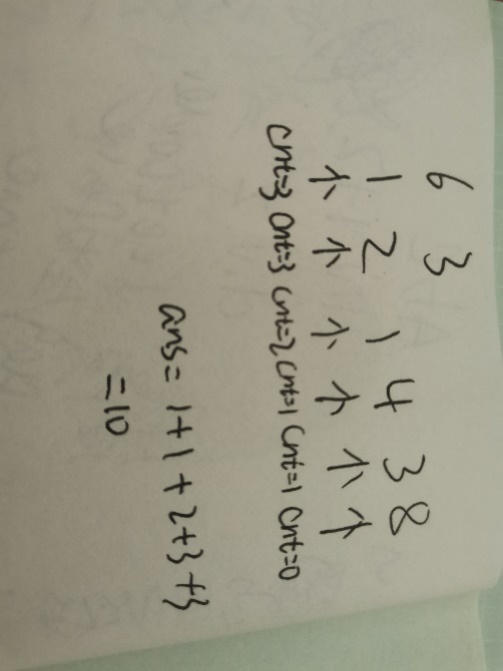
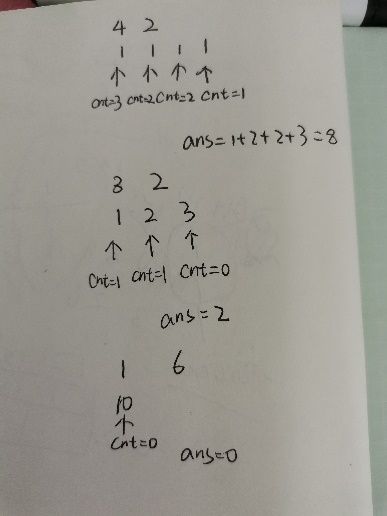
### 原题20

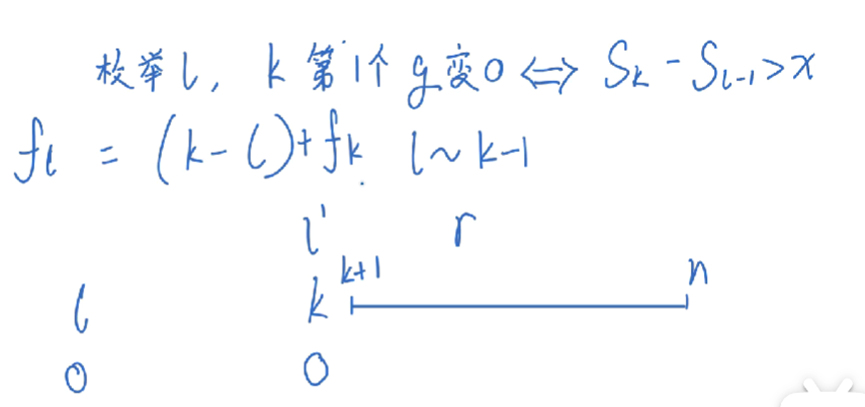
[Problem - B - Codeforces](https://codeforces.com/contest/1994/problem/B)

思路：枚举R，动态规划









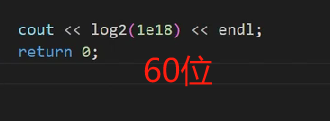
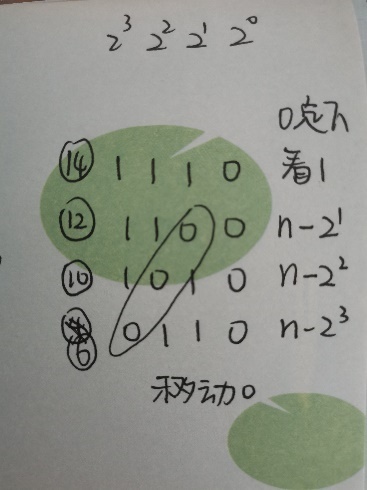
代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. signed main()
5. {
6. int t;
7. cin>>t;
8. while(t--)
9. {
10. int n,x;
11. cin>>n>>x;
12. int a[n];
13. for(int i=0;i<n;i++)
14. cin>>a[i];
15. int now=0,cnt=0,ans=0;
16. for(int i=n-1;i>=0;i--)*//枚举r*
17. {
18. if(now+a[i]<=x)//不超过x
19. {
20. now+=a[i];
21. cnt++;
22. }
23. else{//超过x
24. now=0;
25. }
26. ans+=cnt;//每次都要加cnt
27. }
28. cout<<ans<<endl;
29. }
30. }

### 原题21

[Problem - C - Codeforces](https://codeforces.com/contest/1988/problem/C)

思路：



看n的二进制，0是定下来的，每一位上1，0往左慢慢移动，相当于n-2的i次方

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. signed main()
5. {
6. int t;
7. cin>>t;
8. while(t--)
9. {
10. int n;
11. cin>>n;
12. stack<int> s;//栈的特点：先进后出。先进的大，后进的小，先输出后进
13. s.push(n);
14. for(int i=0;i<64;i++)//n的最大是10的18次方，相当于2的六十次方，多留几位是经验值
15. {
16. if((n>>i)&1)//二进制有1
17. {
18. s.push(n^(1LL<<i));//“-”减号改成“^”异或号，速度快，1往左移动i位相当于2的i次方，LL将1的类型int强制转换为long long,避免越界
19. }
20. }
21. if(s.size()==2)//只有一个1的情况，不能将其改为0，因为题目要求正数
22. {
23. cout<<1<<endl;
24. cout<<n<<endl;
25. continue;
26. }
27. cout<<s.size()<<endl;
28. while(!s.empty())
29. {
30. cout<<s.top()<<" ";
31. s.pop();
32. }
33. cout<<endl;
34. }
35. }

### 原题22

1. [小红装匣子\_牛客周赛 Round 52 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/86373/B)

思路：

2\*a+3\*b=2\*n，所以b必须是偶数，只能上下横着放，a可以竖直放，先填完b,再填a(1\*2,长度为1，很万能)

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. int main()
4. {
5. int t;
6. cin>>t;
7. while(t--)
8. {
9. int a,b,n;
10. cin>>a>>b>>n;
11. int c=min(b/2,n/3);//b的够话，填最多可以填的，b不够的话，按b填
12. if(a>=n-c\*3)
13. cout<<"YES"<<endl;
14. else
15. cout<<"NO"<<endl;
16. }
18. }

### 原题23

1. [小红的数字对对碰\_牛客周赛 Round 52 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/86373/C)

思路：同等正数可消消乐（相同整数异或后为0），负数与正数可消消乐（异或后为负数），负数与负数可消消乐（相加为0），记录相同数用map（因为0~10的9次方）

代码：

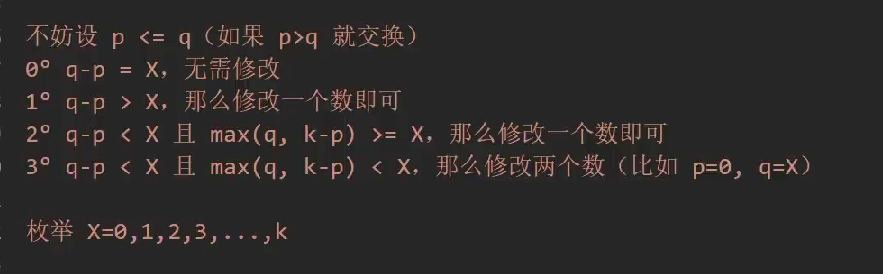
1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. map<int,int> b;
5. signed main()
6. {
7. int n;
8. cin>>n;
9. int a[n],sum1=0,sum2=0;
10. for(int i=0;i<n;i++)
11. {
12. cin>>a[i];
13. if(a[i]>=0)
14. {
15. sum1++;
16. b[a[i]]++;
17. if(b[a[i]]==2)
18. {
19. b[a[i]]-=2;
20. sum1-=2;
21. }
22. }
23. else{
24. sum2++;
25. }
26. }
27. if(sum1>=sum2)*//正数多*
28. {
29. cout<<sum1-sum2;
30. }
31. else{*//负数多*
32. sum2-=sum1;
33. if(sum2%2==0)
34. cout<<0;
35. else
36. cout<<1;
37. }
39. }

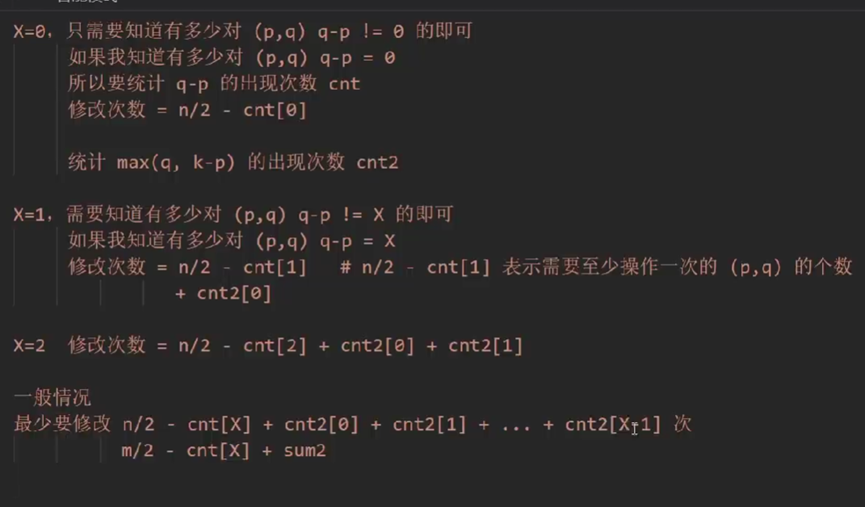
### 原题24

[3224. 使差值相等的最少数组改动次数 - 力扣（LeetCode）](https://leetcode.cn/problems/minimum-array-changes-to-make-differences-equal/description/)

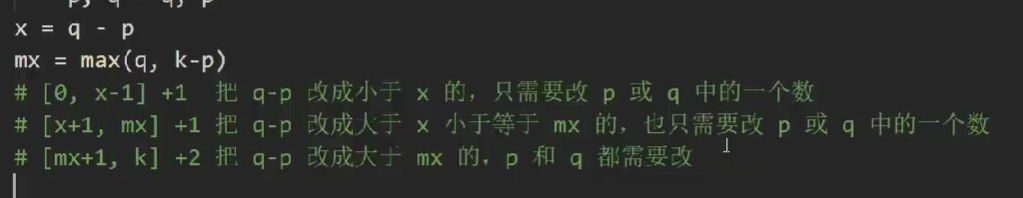
思路：

方法一：枚举





方法二：差分数组



代码：

1. class Solution {
2. public:
3. int minChanges(vector<int>& nums, int k) {
4. vector<int> cnt(k+10),cnt2(k+10);
5. int n=nums.size();
6. for(int i=0;i<n/2;i++)
7. {
8. int p=nums[i],q=nums[n-1-i];
9. cnt[abs(p-q)]++;
10. if(p<q)
11. swap(p,q);
12. cnt2[max(p,k-q)]++;
13. }
14. int ans=1e6,sum2=0;
15. for(int i=0;i<=k;i++)
16. {
17. ans=min(ans,n/2-cnt[i]+sum2);
18. sum2+=cnt2[i];
19. }
20. return ans;
21. }
22. };
23. class Solution {
24. public:
25. int minChanges(vector<int>& nums, int k) {
26. int d[100010];
27. memset(d, 0, sizeof(d));
28. int x,mx;
29. int n=nums.size();
30. for(int i=0;i<n/2;i++)
31. {
32. int p=nums[i],q=nums[n-1-i];
33. if(p<q)
34. swap(p,q);
35. x=p-q;
36. mx=max(p,k-q);
37. d[0]++;
38. d[x]--;
39. d[x+1]++;
40. d[mx+1]--;
41. d[mx+1]+=2;
42. }
43. *//[0,p-q-1]+1*
44. *//[p-q+1,max(p,k-q)]+1*
45. *//[max(p,k-q)+1,k]+2*
46. int ans=n,sum=0;
47. for (int i = 0; i <= k + 1; i++) {
48. sum += d[i];
49. ans = min(ans, sum);
50. }
51. return ans;
52. }
53. };

### 原题25

[3228. 将 1 移动到末尾的最大操作次数 - 力扣（LeetCode）](https://leetcode.cn/problems/maximum-number-of-operations-to-move-ones-to-the-end/description/)

思路：最大操作数，从左边的1开始挪动，记下1的个数，遇到0往右移动

代码：

1. class Solution {
2. public:
3. int maxOperations(string s) {
4. int cnt=0,ans=0;
5. for(int i=0;i<s.size();i++)
6. {
7. if(s[i]=='1')
8. cnt++;
9. else if(s[i]=='0' && i && s[i-1]=='1')
10. ans+=cnt;
11. }
12. return ans;
13. }
14. };

### 原题26

[3229. 使数组等于目标数组所需的最少操作次数 - 力扣（LeetCode）](https://leetcode.cn/problems/minimum-operations-to-make-array-equal-to-target/description/)

思路：差分数组，多少个正数对应多少个负数，计算出相应的正数或负数即可

如样例二，前缀和数组为[1，-2，2]，答案为5，差分数组为[1，-3，4，（-2）],计算正数之和或负数之和即可

代码：

1. class Solution {
2. public:
3. long long minimumOperations(vector<int>& nums, vector<int>& target) {
4. int n=nums.size();
5. vector<int> c(n);//前缀和
6. vector<int> d(n);//差分数组
7. for(int i=0;i<n;i++)
8. {
9. c[i]=target[i]-nums[i];
10. }
11. for(int i=n-1;i>0;i--)
12. {
13. d[i]=c[i]-c[i-1];
14. }
15. d[0]=c[0];
16. long long cnt=0,cnt2=0;
17. for(int i=0;i<n;i++)
18. {
19. if(d[i]>0)
20. cnt+=d[i];
21. else
22. cnt2-=d[i];
23. }
24. return max(cnt,cnt2);
25. }
26. };

### 原题27

[Problem - A - Codeforces](https://codeforces.com/contest/1990/problem/A)

思路：贪心，只要有一种数是奇数即可赢

### 原题28

[Problem - B - Codeforces](https://codeforces.com/contest/1990/problem/B)

思路：贪心，1，2，3，…y，…，x，……

Y,X中间全赋值为1，左右两边尽量平衡，-1，1，-1的交替

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. int main()
4. {
5. int t;
6. cin>>t;
7. while(t--)
8. {
9. int n,x,y;
10. cin>>n>>x>>y;
11. int a[n+1];
12. for(int i=y;i<=x;i++)
13. a[i]=1;
14. int c=0;
15. for(int i=y-1;i>=1;i--)
16. {
17. if(c%2==0)
18. {
19. a[i]=-1;
20. c++;
21. }
22. else
23. {
24. a[i]=1;
25. c++;
26. }
27. }
28. int d=0;
29. for(int i=x+1;i<=n;i++)
30. {
31. if(d%2==0)
32. {
33. a[i]=-1;
34. d++;
35. }
36. else
37. {
38. a[i]=1;
39. d++;
40. }
41. }
42. for(int i=1;i<=n;i++)
43. cout<<a[i]<<" ";
44. cout<<endl;
45. }
46. }

### 原题29

[Problem - B1 - Codeforces](https://codeforces.com/contest/1995/problem/B1)

思路：双指针

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. signed main()
5. {
6. int t;
7. cin>>t;
8. while(t--)
9. {
10. int n,m;
11. cin>>n>>m;
12. int a[n];
13. for(int i=0;i<n;i++)
14. cin>>a[i];
15. sort(a,a+n);
16. int sum=0,ans=0;
17. for(int l=0,r=0;r<n;r++)
18. {
19. sum+=a[r];
20. while(l<=r && (sum>m || abs(a[l]-a[r])>1))
21. {
22. sum-=a[l];
23. l++;
24. }
25. ans=max(ans,sum);
26. }
27. cout<<ans<<endl;
28. }
29. }

### 原题30

[Problem - B2 - Codeforces](https://codeforces.com/contest/1995/problem/B2)

思路：贪心

每个数都不一样，差值不大于1，只能选两个数。

极限思想：每个数单独都算一遍

混合版：先贪心选小的数，再选大的数，再考虑在小的数可以填补多少（+1）

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. signed main()
5. {
6. int t;
7. cin>>t;
8. while(t--)
9. {
10. int n,m;
11. cin>>n>>m;
12. vector< pair<int,int> > a(n);
13. for(int i=0;i<n;i++)
14. cin>>a[i].first;
15. for(int i=0;i<n;i++)
16. cin>>a[i].second;
17. sort(a.begin(),a.end());
18. int ans=0,cnt;
19. for(int i=0;i<n;i++)
20. {
21. cnt=min(a[i].second,m/a[i].first);
22. ans=max(ans,a[i].first\*cnt);
23. }
24. for(int i=1;i<n;i++)
25. {
26. if(abs(a[i].first-a[i-1].first)>1)
27. continue;
28. int cur=m;
29. int small=min(a[i-1].second,cur/a[i-1].first);
30. cur-=small\*a[i-1].first;
31. int big=min(a[i].second,cur/a[i].first);
32. cur-=big\*a[i].first;
33. int add=min(min(small,a[i].second-big),cur);
34. cur-=add;
35. ans=max(ans,m-cur);
36. }
37. cout<<ans<<endl;
38. }
39. }

### 原题31

[统计不是特殊数字的数字数量 - 力扣（LeetCode）](https://leetcode.cn/contest/weekly-contest-408/problems/find-the-count-of-numbers-which-are-not-special/description/)

思路：前缀和

代码：

1. class Solution {
2. private:
3. int is\_prime(int x)
4. {
5. for(int i=2;i\*i<=x;i++)
6. {
7. if(x%i==0)
8. return 0;
9. }
10. return 1;
11. }
12. int cnt(int x)*//时间复杂度为O(10000 \* 100 == 10的6次方)*
13. {
14. int ans=0;
15. for(int i=2;i\*i<=x;i++)
16. {
17. if(is\_prime(i))
18. ans++;
19. }
20. return ans;
21. }
22. public:
23. int nonSpecialCount(int l, int r) {
24. return r-l+1-(cnt(r)-cnt(l-1));
25. }
26. };

### 原题32

[Problem - C - Codeforces](https://codeforces.com/contest/1996/problem/C)

思路：前缀和，统计26个字母的区间数量，预处理，查询快

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. int main()
4. {
5. int t;
6. cin>>t;
7. while(t--)
8. {
9. int n,q;
10. cin>>n>>q;
11. int cnt1[n+10][50],cnt2[n+10][50];
12. memset(cnt1,0,sizeof(cnt1)),memset(cnt2,0,sizeof(cnt2));
13. string a,b;
14. cin>>a>>b;
15. cnt1[0][a[0]-'a']++;
16. cnt2[0][b[0]-'a']++;
17. for(int i=1;i<n;i++)
18. {
19. for(int j=0;j<26;j++)
20. {
21. cnt1[i][j]=cnt1[i-1][j];
22. }
23. cnt1[i][a[i]-'a']++;
24. }
25. for(int i=1;i<n;i++)
26. {
27. for(int j=0;j<26;j++)
28. {
29. cnt2[i][j]=cnt2[i-1][j];
30. }
31. cnt2[i][b[i]-'a']++;
32. }
33. while(q--)
34. {
35. int l,r;
36. cin>>l>>r;
37. int sum1=0;
38. l=l-1,r=r-1;
39. for(int i=0;i<26;i++)
40. {
41. if(l)sum1+=abs(cnt1[r][i]-cnt1[l-1][i]-(cnt2[r][i]-cnt2[l-1][i]));
42. else sum1+=abs(cnt1[r][i]-cnt2[r][i]);
43. }
44. cout<<sum1/2<<endl;
45. }
47. }
48. }

### 原题33

[3234. 统计 1 显著的字符串的数量 - 力扣（LeetCode）](https://leetcode.cn/problems/count-the-number-of-substrings-with-dominant-ones/description/)

思路：枚举

代码：

1. using namespace std;
2. class Solution {
3. public:
4. int numberOfSubstrings(string s) {
5. vector<int> a;//记录字符串中0的下标
6. for(int i=0;i<s.size();i++)
7. {
8. if(s[i]=='0')
9. a.push\_back(i);
10. }
11. int tol=s.size()-a.size();//1的总数
12. a.push\_back(s.size());//添加哨兵n
13. int ans=0;int k=0;
14. for(int l=0;l<s.size();l++)//枚举左端点
15. {
16. for(;k<a.size();k++)//在左端点的右边第一个0
17. {
18. if(a[k]>=l)
19. break;
20. }
21. ans+=a[k]-l;
22. for (int i=k;i<a.size()-1;i++)//枚举0的下标
23. {
24. int p=a[i];
25. int q=a[i+1];
26. int C0=i-k+1;
27. if(C0 \*C0 > tol)break;
28. int C1=a[i]-l-(i-k);
29. if(C0 \* C0 <= C1)
30. {
31. ans += q-p;
32. }
33. else
34. {
35. ans += max(q-p-(C0 \* C0 - C1),0);
36. }//缺多少往前跳多少才够
37. }
39. }
40. return ans;
41. }
42. };

### 原题34

[Problem - B - Codeforces](https://codeforces.com/contest/1991/problem/B)

思路：|或运算，&运算，有1填上1

代码：

1. #include<bits/stdc++.h>
2. using namespace std;
3. #define int long long
4. signed main()
5. {
6. int t;
7. cin>>t;
8. while(t--)
9. {
10. int n;
11. cin>>n;
12. int b[n];
13. for(int i=0;i<n-1;i++)
14. cin>>b[i];
15. int a[n];
16. memset(a,0,sizeof a);
17. for(int i=0;i<n-1;i++)
18. {
19. for(int j=0;j<=30;j++)
20. {
21. if(b[i]>>j&1)
22. {
23. a[i] |= 1LL<<j;
24. a[i+1] |= 1LL<<j;
25. }
26. }
27. }
28. int flag=0;
29. for(int i=0;i<n-1;i++)
30. {
31. if((a[i] & a[i+1]) != b[i])
32. {
33. cout<<"-1"<<endl;
34. flag=1;
35. break;
36. }
37. }
38. if(flag==1)continue;
39. for(int i=0;i<n;i++)
40. {
41. cout<<a[i]<<" ";
42. }
43. cout<<endl;
44. }
46. }

### 原题35

[Problem - C - Codeforces](https://codeforces.com/contest/2001/problem/C)

思路：并查集，二分，每问两个点，给出的答案是中点，若是奇数，给离的（问1）点最近的那个

。直到问到的答案是（问1）点那就是，问的两点是父子关系

代码：

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  const int N=2000;  int fa[N];  signed main()  {      int t;      cin>>t;      while(t--)      {          int n;          cin>>n;          vector<int> ans;          for(int i=2;i<=n;i++)          {              cout<<"? 1 "<<i<<endl;              cout.flush();              int a;              cin>>a;              if(a==1)              {                  fa[i]=1;              }              else{                  while(a!=i)                  {                      fa[i]=a;                      cout<<"? "<<i<<" "<<a<<endl;                      cout.flush();                      cin>>a;                  }              }          }          cout<<"! ";          for(int i=2;i<=n;i++)          {              cout<<fa[i]<<" "<<i<<" ";          }          cout<<endl;      }  } |

### 原题36

[C-迷宫\_牛客小白月赛99 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/88455/C)

思路：

法1：将S能到达的地区标记为1，E能到达的地区标记为2，然后判定每个2的附近3行行列是否有1，有的话就可以直接贯穿

法2：起点与附近形成一个连通块，终点与附近形成一个连通块。从终点连通块入手，每个点的3行3列标记了，再从起点连通块入手，有标记过的就是能到达

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **const** **int** N=1010;  **int** n,m,s1,s2,e1,e2;  **char** a[N][N];  **int** vis[N][N],stx[N],sty[N];  **int** dx[4]={1,-1,0,0},dy[4]={0,0,1,-1};    **void** solve()  {      cin>>n>>m;  **for**(**int** i=1;i<=n;i++)      {  **for**(**int** j=1;j<=m;j++)          {              cin>>a[i][j];  **if**(a[i][j]=='S')              {                  s1=i,s2=j;              }  **if**(a[i][j]=='E')              {                  e1=i,e2=j;              }          }      }        queue< pair<**int**,**int**> > q;      q.push({e1,e2});  **while**(q.size())      {  **int** x=q.front().first,y=q.front().second;          q.pop();    **if**(x<1||x>n||y<1||y>m||vis[x][y])**continue**;          vis[x][y]=2;          stx[x+1]=stx[x]=stx[x-1]=1;          sty[y+1]=sty[y]=sty[y-1]=1;    **for**(**int** i=0;i<4;i++)          {  **int** newx=x+dx[i];  **int** newy=y+dy[i];  **if**(newx<1||newx>n||newy<1||newy>m)**continue**;  **if**(a[newx][newy]!='#' && !vis[newx][newy])              {                  q.push({newx,newy});              }          }      }    **int** flag=(vis[s1][s2]==2);      q.push({s1,s2});  **while**(q.size())      {  **int** x=q.front().first,y=q.front().second;          q.pop();        **if**(vis[x][y]==1)**continue**;          vis[x][y]=1;    **if**(stx[x] || sty[y])              flag=1;    **for**(**int** i=0;i<4;i++)          {  **int** newx=x+dx[i];  **int** newy=y+dy[i];  **if**(newx<1||newx>n||newy<1||newy>m)**continue**;  **if**(vis[newx][newy]==2)                  flag=1;  **if**(a[newx][newy]!='#' && vis[newx][newy]!=1)              {                  q.push({newx,newy});              }          }      }    **if**(flag)          cout<<"YES"<<endl;  **else**          cout<<"NO"<<endl;  }    **signed** main(){        solve();  **return** 0;  } |

### 原题37

[C-最大区间和\_长沙学院2024年程序设计竞赛春季校赛 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/84832/C)

思路：预处理，前缀和，贪心，最大值=最大值-最小值（0或负的）

0 <= i <= l-1 ,Pre[i]为区间sum最小值（0或负的）

R <= i <=n ,suf[i]为区间sum最大值

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **const** **int** N=200010;  **int** a[N],sum[N],pre[N],suf[N];  **signed** main()  {  **int** n,q;      cin>>n>>q;  **for**(**int** i=1;i<=n;i++)          cin>>a[i];      //预处理sum[i]  **for**(**int** i=1;i<=n;i++)          sum[i]=sum[i-1]+a[i];      //预处理前缀最小sum  **for**(**int** i=1;i<=n;i++)      {          pre[i]=min(pre[i-1],sum[i]);      }      //预处理后缀最大sum      suf[n]=sum[n];  **for**(**int** i=n-1;i>=1;i--)      {          suf[i]=max(suf[i+1],sum[i]);      }  **while**(q--)      {  **int** l,r;          cin>>l>>r;          cout<<suf[r]-pre[l-1]<<endl;      }  } |

### 原题38

[Problem - F - Codeforces](https://codeforces.com/contest/1992/problem/F)

思路：集合维护约数，不能出现集合中的数，否则割掉

代码：

|  |
| --- |
| #include <iostream>  #include <set>  using namespace std;  signed main(){      int t;      cin >> t;      for(int i = 0; i < t; i++){          int n, x;          cin >> n >> x;          int cnt = 0;          set<int> avoid;          avoid.insert(x);          for(int j = 0; j < n; j++){              int a;              cin >> a;              if(avoid.count(a)){                  avoid.clear();                  avoid.insert(x);                  cnt++;              }              for(auto y: avoid){                  if(y%a == 0){                      avoid.insert(y/a);                  }              }          }          cout << cnt + 1 << endl;      }      return 0;  } |

### 原题39

[B-tb的字符串问题\_牛客小白月赛101 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/90072/B)

思路：括号匹配问题，用栈来实现

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**signed** main()

{

**int** n;

    cin>>n;

    string s;

    cin>>s;

    stack<**char**> stk;

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

    {

        stk.push(s[i]);

**if**(stk.size()>1)

        {

**int** p1=stk.top();

            stk.pop();

**int** p2=stk.top();

            stk.pop();

**if**(p1=='b'&&p2=='t' || p1=='c'&&p2=='f')

            {

**continue**;

            }

            stk.push(p2);

            stk.push(p1);

        }

    }

    cout<<stk.size();

}

### 原题40

[C - Separated Lunch (atcoder.jp)](https://atcoder.jp/contests/abc374/tasks/abc374_c)

思路：枚举，每个部门的状态

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

int n,a[100],vis[100],sum=0,ans=0x3f3f3f3f;

void dfs(int k)

{

    if(k>n)

    {

        int mid=0;

        for(int i=1;i<=n;i++)

        {

            if(vis[i])

                mid+=a[i];

        }

        ans=min(ans,max(mid,sum-mid));

        return;

    }

    vis[k]=1;

    dfs(k+1);

    vis[k]=0;

    dfs(k+1);

}

signed main()

{

    cin>>n;

    for(int i=1;i<=n;i++)

    {

        cin>>a[i];

        sum+=a[i];

    }

    dfs(1);

    cout<<ans;

 }

### 原题41

[Problem - A - Codeforces](https://codeforces.com/contest/2023/problem/A)

思路：贪心，比较难证明，要有经验

用pair存起来，不能改变顺序，将pair数组串联起来，

和之比

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

bool cmp(pair<int,int> a,pair<int,int> b)

{

    return a.first+a.second < b.first+b.second;

 }

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        pair<int,int> p[n];

        for(int i=0;i<n;i++)

        {

            cin>>p[i].first>>p[i].second;

        }

        sort(p,p+n,cmp);

        for(int i=0;i<n;i++)

        {

            cout<<p[i].first<<" "<<p[i].second<<" ";

        }

        cout<<endl;

    }

}

### 原题42

[D-小红的数位删除\_第20届纪念款-牛客周赛 Round 20 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/69695/D)

思路：

/\*

基本思路就是

for(枚举数字a的二进制状态) {

获取a在此状态下变成的数字x，并记录保留了几位数字（后续用于计算删除了几位数字

for(枚举数字b的二进制状态) {

获取b在此状态下变成的数字y，并记录保留了几位数字（后续用于计算删除了几位数字

if(满足谁是谁的倍数) {

更新答案

}

}

}

\*/

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**signed** main()

{

    string a,b;

    cin>>a>>b;

**int** la=a.size(),lb=b.size();

**int** ans=60001000;

**for**(**int** i=1;i<(1<<la);i++)//用二进制表示a的位数上是否保留，1是0否

    {

**int** x=0;

**int** cnt1=0;

**for**(**int** j=0;j<la;j++)

        {

**if**(i>>j&1)

                x=x\*10+a[j]-'0',cnt1++;

        }

**for**(**int** ii=1;ii<(1<<lb);ii++)

        {

**int** y=0,cnt2=0;

**for**(**int** jj=0;jj<lb;jj++)

            {

**if**(ii>>jj&1)

                    y=y\*10+b[jj]-'0',cnt2++;

            }

**if**(x%y==0 || y%x==0)

            {

                ans=min(ans,la+lb-cnt1-cnt2);

            }

        }

    }

**if**(ans==60001000)

        cout<<"-1";

**else**

        cout<<ans;

}

### 原题43

[I-小A取石子\_牛客小白月赛13 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/549/I)

思路：简单博弈论，^代表每一堆石头的二进制状态是奇数个1还是偶数个1，奇数先手必胜，偶数先手必败

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**signed** main()

{

**int** n,k;

    cin>>n>>k;

**int** a[n];

**int** ans=0;

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

    {

        cin>>a[i];

        ans ^= a[i];

    }

**if**(0<k && k<=n || ans!=0)

        cout<<"YES";

**else**

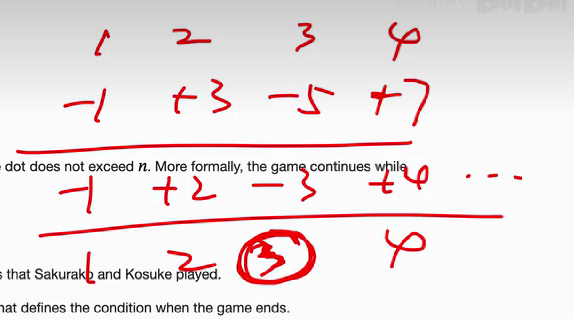
        cout<<"NO";

}

### 原题44

[Problem - A - Codeforces](https://codeforces.com/contest/2033/problem/A)

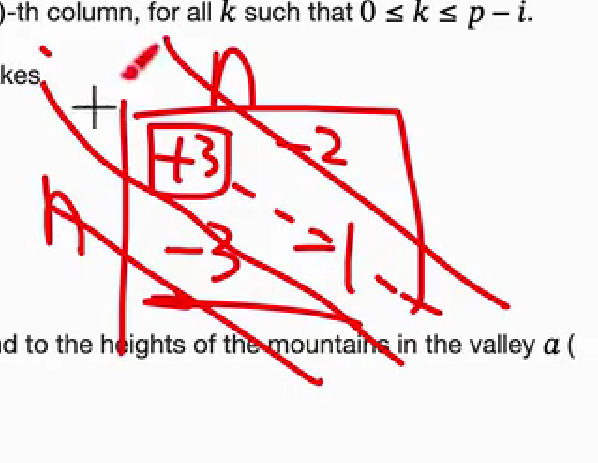
思路：规律题，奇数偶数

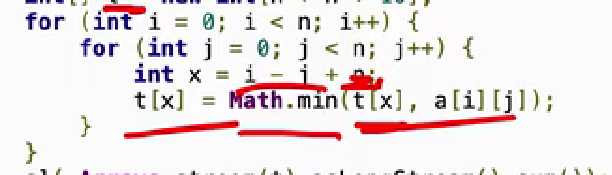


### 原题45

[Problem - B - Codeforces](https://codeforces.com/contest/2033/problem/B)

思路：主对角线，i-j+n



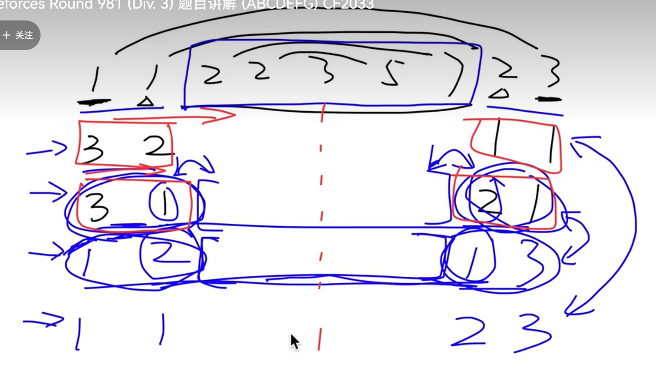


2\*n-1条对角线

### 原题46

[Problem - C - Codeforces](https://codeforces.com/contest/2033/problem/C)

思路：



最优解，两种状态：不换（全换），换一半

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        int a[n];

        for(int i=0;i<n;i++)

            cin>>a[i];

        for(int i=1;i<n/2;i++)

        {

            int t1=0;

            int t2=0;

            if(a[i]==a[i-1])

                t1++;

            if(a[n-i-1]==a[n-i])

                t1++;

            if(a[n-i-1]==a[i-1])

                t2++;

            if(a[n-i]==a[i])

                t2++;

            if(t1>t2)

            {

                int temp=a[i];

                a[i]=a[n-i-1];

                a[n-i-1]=temp;

            }

        }

        int ans=0;

        for(int i=1;i<n;i++)

        {

            if(a[i]==a[i-1])

                ans++;

        }

        cout<<ans<<endl;

    }

 }

### 原题47

[取石子 (nowcoder.com)](https://ac.nowcoder.com/acm/problem/15972)

思路：简单博弈论，只能取奇数个石头，一开始两堆石头之和为奇数，先手可以取一点，使其变为偶数，这样后手就不可能完全取完。反之

奇数，先手必胜；

偶数，后手必胜。

### 原题48

[Problem - D - Codeforces](https://codeforces.com/contest/2033/problem/D)

思路：求不重叠最多线段（线段和为0），用前缀和，若当前缀和出现与之前的前缀和一样，说明中间的线段为0。特判一开始前缀和为0。

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        int a[n],sum[n];

        for(int i=0;i<n;i++)

            cin>>a[i];

        sum[0]=a[0];

        for(int i=1;i<n;i++)

            sum[i]=sum[i-1]+a[i];

        int ans=0;

        set<int> s;

        s.insert(0);

        for(int i=0;i<n;i++)

        {

            if(s.count(sum[i]))

            {

                ans++;

                s.clear();

            }

            s.insert(sum[i]);

        }

        cout<<ans<<endl;

    }

}

### 原题49

[栗酱的异或和 (nowcoder.com)](https://ac.nowcoder.com/acm/problem/14619)

思路：先手从第k堆开始取，若取完后，剩下的异或和为0，后手将面临必败态。  
所以先计算除了a[k]外的其他数字的异或和，如果a[k]大于这个sum，则我们取掉sum-a[k]即可。

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**signed** main()

{

**int** t;

    cin>>t;

**while**(t--)

    {

**int** n,k;

        cin>>n>>k;

**int** a[n];

**int** res=0,ak;

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

        {

            cin>>a[i];

**if**(i==k-1)

                ak=a[k-1];

**else**

                res^=a[i];

        }

**if**(res < ak)

                cout<<"Yes"<<endl;

**else**

                cout<<"No"<<endl;

    }

}

### 原题50

[Problem - B - Codeforces](https://codeforces.com/contest/2027/problem/B)

思路：删掉一些数字，变成脆弱数组。脆弱数组就是第一个数字比后面的等于或大。（非递增）

代码：

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        int a[n];

        for(int i=0;i<n;i++)

        {

            cin>>a[i];

        }

        int res=0;

        for(int i=0;i<n;i++)

        {

            int cnt=0;

            for(int j=i;j<n;j++)

            {

                if(a[i]>=a[j])

                    cnt++;

            }

            res=max(res,cnt);

        }

        cout<<n-res<<endl;

    }

}

### 原题51

[Problem - C - Codeforces](https://codeforces.com/contest/2027/problem/C)

思路： 搜索，找最优解

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

map<int,vector<int>> mp1; //mp1[a[i]+i-1]=贡献值 (贡献值唯一,因为i-1唯一)

map<int,int> mp2; //标记数组 mp2[贡献值]=1

int res;

void dfs(int number)

{

    res=max(res,number);

    if(!mp1[number].empty())

    {

        for(auto i:mp1[number])

        {

            if(mp2[i])

                continue;

            mp2[i]=1;

            dfs(number+i);

        }

    }

}

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        mp1.clear();

        mp2.clear();

        int n;

        cin>>n;

        int a[n+10];

        for(int i=1;i<=n;i++)

        {

            cin>>a[i];

        }

        for(int i=2;i<=n;i++)

        {

            mp1[a[i]+i-1].push\_back(i-1);

        }

        res=n;

        dfs(n);

        cout<<res<<endl;

    }

}

### 原题52

[D-医生\_牛客周赛 Round 65 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/92972/D)

思路：状态压缩，把每一种药物选不选的可能枚举出来，得出治疗效果，若治疗效果可以治愈病人，就比较是不是最优解。若都没有，就是-1.

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**int** a[10005], b[15], c[1024];

**signed** main()

{

**int** n,m;

    cin>>n>>m;

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

    {

        string s;

        cin>>s;

        a[i]=stoi(s,0,2);

    }

**int** k;

    cin>>k;

**for**(**int** i=0;i<k;i++)

    {

        string s;

        cin>>s;

        b[i]=stoi(s,0,2);

    }

    map<**int**,**int**> mp;//状态上有多少个1

    //c[状态]=可治疗功效,状态压缩，k种药物，1~2的k次方-1  (1代表选选这种药物，0代表不选这种药物)

**for**(**int** i=0;i<(1<<k);i++)//枚举每一种状态

    {

        //枚举状态上的每一位，1代表选该位置的药物

**for**(**int** j=0;j<k;j++)

        {

**if**(i>>j&1)

            {

                c[i] |= b[j];

                mp[i]++;

            }

        }

    }

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

    {

**int** ans=k+1;

**for**(**int** j=0;j<(1<<k);j++)///枚举c

        {

**if**((c[j]&a[i])==a[i])

            {

                ans=min(ans,mp[j]);

            }

        }

**if**(ans==k+1)

            cout<<-1<<endl;

**else**

            cout<<ans<<endl;

    }

}

### 原题53

[D-冰冰的分界线\_牛客小白月赛103 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/93218/D)

思路：中垂线，到两点距离一样，y=k\*x+b

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**const** **int** mod=1000000007;

//乘法逆元，求出k,b（有取模操作，k,b唯一）

**int** qmi(**int** a, **int** b, **int** p)

{

**int** res = 1;

**while** (b)

    {

**if** (b & 1) res = res \* a % p;

        a = a \* (**int**)a % p;

        b >>= 1;

    }

**return** res;

}

**signed** main()

{

**int** t;

    cin>>t;

**while**(t--)

    {

        set< pair<**int**,**int**> > s;

**int** n;

        cin>>n;

**int** x[n],y[n];

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

            cin>>x[i];

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

            cin>>y[i];

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

        {

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

            {

**int** k;

**if**(y[i]==y[j])

                    k=0x3f3f;

**else** **if**(x[i]==x[j])

                    k=0;

**else**

                    k=(x[i]-x[j])\*qmi(y[j]-y[i],mod-2,mod)%mod;

**int** b=(y[j]\*y[j]-y[i]\*y[i]+x[j]\*x[j]-x[i]\*x[i])\*qmi(2\*(y[j]-y[i]),mod-2,mod)%mod;

**if**(k==0x3f3f)

                {

                    s.insert({k,x[i]+x[j]});

                }

**else**{

                    s.insert({k,b});

                }

            }

        }

        cout<<s.size()<<endl;

    }

}

### 原题54

[E-降温（easy）\_牛客周赛 Round 65 (nowcoder.com)](https://ac.nowcoder.com/acm/contest/92972/E)

思路：贪心

最大：  
如果b[i-1]-x < -50，反正b[i]已经不是寒潮了，直接将b[i]设为50，为后续的寒潮做准备  
否则将b[i]设为b[i-1]-x，以便在保证当前是一个寒潮的情况下，为后续的寒潮留有尽量多的空间  
最小：  
将b[i]刚好设为b[i-1]-x+1，在保证当前不是寒潮的情况下让温度尽量低，当然最低只能为-50

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**const** **int** N=1024;

**signed** main()

{

**int** n,x;

    cin>>n>>x;

**int** a[N];

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

    {

        cin>>a[i];

    }

**int** a1[N],a2[N];

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

    {

        a1[i]=a[i];

    }

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

    {

        a2[i]=a[i];

    }

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

    {

**if**(a1[i]==-999)

        {

**if**(i==0)

            {

                a1[i]=50;

            }

**else** **if**(a1[i-1]-x<-50)

            {

                a1[i]=50;

            }

**else**{

                a1[i]=a1[i-1]-x;

            }

        }

    }

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

    {

**if**(a2[i]==-999)

        {

**if**(i==0)

            {

                a2[i]=-50;

            }

**else** **if**(a2[i-1]-x+1<-50)

            {

                a2[i]=-50;

            }

**else**

                a2[i]=a2[i-1]-x+1;

        }

    }

**int** ans1=0,ans2=0;

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

    {

**if**(a1[i-1]-a1[i]>=x)

        {

            ans1++;

        }

    }

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

    {

**if**(a2[i-1]-a2[i]>=x)

        {

            ans2++;

        }

    }

    cout<<ans1<<" "<<ans2<<endl;

}

### 原题54

[Problem - B - Codeforces](https://codeforces.com/contest/2032/problem/B)

思路：构造题目，中位数的中位数是答案

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n,k;

        cin>>n>>k;

        if(n==1 && k==1)

        {

            cout<<1<<endl;

            cout<<1<<endl;

        }

        else{

            if(n<k || k==1 || k==n || n%2==0)

                cout<<-1<<endl;

            else{

                cout<<3<<endl;

                int a=1,b,c;

                if((n-k)%2==0)

                {

                    b=k-1,c=k+2;

                }

                else

                    b=k,c=k+1;

                cout<<a<<" "<<b<<" "<<c<<" "<<endl;

            }

        }

    }

}

### 原题55

[Problem - D - Codeforces](https://codeforces.com/contest/2036/problem/D)

思路：顺时针

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n,m;

        cin>>n>>m;

        int a[n+10][m+10];

        for(int i=0;i<n+10;i++)

        {

            for(int j=0;j<m+10;j++)

            {

                a[i][j]=-1;

            }

        }

        int b[n+10][m+10];

        getchar();

        for(int i=0;i<n;i++)

        {

            for(int j=0;j<m;j++)

            {

                char ch;

                cin>>ch;

                a[i][j]=ch-'0';

                b[i][j]=1;

            }

            getchar();

        }

        int N=0,s=n\*m;

        int i=0,j=0;

        int res=0;

        while(N<s)

        {

            vector<int> ans;

            while(a[i][j]!=-1&&b[i][j])//→

            {

                b[i][j]=0;

                ans.push\_back(a[i][j]);

                j++;

                N++;

            }

            i++;j--;

            while(a[i][j]!=-1&&b[i][j])//↓

            {

                b[i][j]=0;

                ans.push\_back(a[i][j]);

                i++;

                N++;

            }

            i--;j--;

            while(a[i][j]!=-1&&b[i][j])//←

            {

                b[i][j]=0;

                ans.push\_back(a[i][j]);

                j--;

                N++;

            }

            i--;j++;

            while(a[i][j]!=-1&&b[i][j])//↑

            {

                b[i][j]=0;

                ans.push\_back(a[i][j]);

                i--;

                N++;

            }

            i++;j++;

            int mod=2\*(n+m)-4;//一个周期

            for(int g=0;g<mod;g++)

            {

                if(ans[g]==1 && ans[(g+1)%mod]==5 && ans[(g+2)%mod]==4 && ans[(g+3)%mod]==3)

                {

                    res++;

                }

            }

            n-=2;

            m-=2;

        }

        cout<<res<<endl;

    }

}

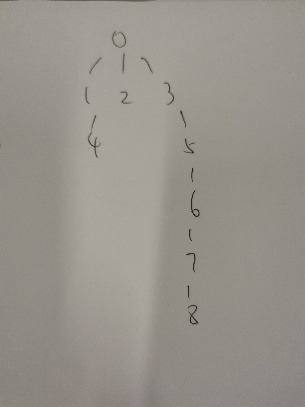
### 原题54

[Problem - D - Codeforces](https://codeforces.com/contest/2032/problem/D)

思路：交互题目

是有顺序的

有层次关系，用bfs队列



代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

const int N=100010;

int ask(int a,int b)

{

    cout<<"? "<<a<<" "<<b<<endl;

    int res;

    cin>>res;

    return res;

}

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        //初始化

        int f[N];//0~n-1节点的父亲

        f[1]=0;//1的父节点必须是0

        queue<int> q;//正在找儿子的节点

        q.push(1);

        int now=2;//找父亲

        while(now<n)//0

        {

            int res=ask(1,now);

            if(res==1)//now的父节点是0

            {

                f[now]=0;

                q.push(now);

                now++;

            }

            else

            {

                f[now]=1;

                q.pop();

                q.push(now);

                now++;

                break;

            }

        }

        while(now<n)

        {

            int res=ask(q.front(),now);

            if(res==0)//不含0，才能确定

            {

                f[now]=q.front();

                q.push(now);

                now++;

            }

            q.pop();

        }

        cout<<"! ";

        for(int i=1;i<n;i++)

        {

            cout<<f[i]<<" ";

        }

        cout<<endl;

    }

}

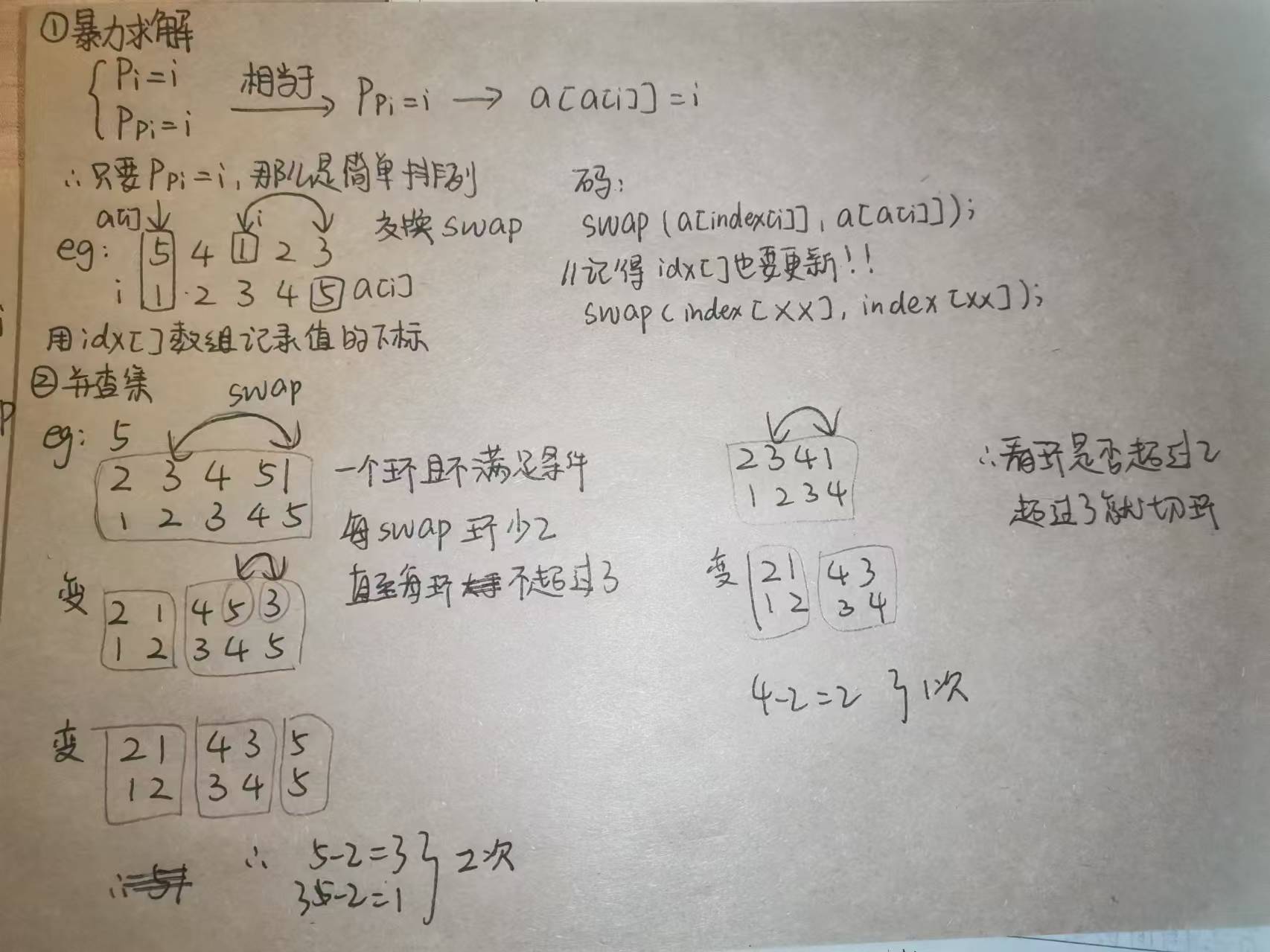
### 原题54

[Problem - E - Codeforces](https://codeforces.com/contest/2033/problem/E)

思路：

1.暴力：

2.并查集



代码：

1.

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        int a[n+10];

        int index[n+10];

        for(int i=1;i<=n;i++)

        {

            cin>>a[i];

            index[a[i]]=i;

        }

        int res=0;

        for(int i=1;i<=n;i++)

        {

            if(a[a[i]]!=i)

            {

                res++;

                swap(a[index[i]],a[a[i]]);

                swap(index[a[a[i]]],index[a[index[i]]]);

            }

        }

        cout<<res<<endl;

    }

}

2.

#include<bits/stdc++.h>

using namespace std;

#define int long long

const int N=1000100;

int p[N];

int find(int x)

{

    if(x!=p[x])

        p[x]=find(p[x]);

    return p[x];

}

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        int a[n+10];

        for(int i=1;i<=n;i++)

        {

            p[i]=i;

        }

        for(int i=1;i<=n;i++)

        {

            cin>>a[i];

            p[find(a[i])]=find(i);

        }

        set<int> s;

        for(int i=1;i<=n;i++)

        {

            if(s.find(find(i))==s.end())

            {

                s.insert(find(i));

            }

        }

        map<int,int> mp;

        for(int i=1;i<=n;i++)

        {

            mp[find(a[i])]++;

        }

        int ans=0;

        for(auto j:s)

        {

            if(mp[j]<=2)

                continue;

            while(mp[j]>2)

            {

                mp[j]-=2;

                ans++;

            }

        }

        cout<<ans<<endl;

    }

}

### 原题55

[Problem - A - Codeforces](https://codeforces.com/contest/2031/problem/A)

思路：

贪心，找最多的数字次数，然后把其它数字改了，保证递增，这样就是最少的。

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        int a[n];

        map<int,int> mp;

        int ma=0;

        for(int i=0;i<n;i++)

        {

            cin>>a[i];

            mp[a[i]]++;

            ma=max(ma,mp[a[i]]);

        }

        cout<<n-ma<<endl;

    }

 }

Codeforces 奇数偶数，数字的次数

暴力枚举

### 原题56

[Problem - B - Codeforces](https://codeforces.com/contest/2031/problem/B)

思路：暴力，一重循环，判断符合条件不，符合就执行操作

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        int a[n+10];

        for(int i=1;i<=n;i++)

        {

            cin>>a[i];

        }

        int flag=1;

        for(int i=1;i<=n;i++)

        {

            if(a[i]==i)

                continue;

            if(a[i+1]==i && abs(a[i+1]-a[i])==1)

            {

                swap(a[i+1],a[i]);

                continue;

            }

            flag=0;

        }

        if(flag)

            cout<<"Yes"<<endl;

        else

            cout<<"No"<<endl;

    }

 }

Codeforces 构造题，找到最基本情况来构造，再特判

### 原题57

[Problem - C - Codeforces](https://codeforces.com/contest/2031/problem/C)

思路：构造题，偶数必有，1 1 2 2 3 3……向上取整

奇数根据勾股数3,4,5（9+16=25），来填三个数(坐标为1 10 26，填1)，2~9是偶数个，11~25是奇数个，所以拿出个23配27(4)，剩下偶数个，妥妥的。用map当计数器

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

    int t;

    cin>>t;

    while(t--)

    {

        int n;

        cin>>n;

        if(n%2==0)//ż������:1 1 2 2 3 3...

        {

            for(int i=1;i<=n;i++)

            {

                cout<<(i+1)/2<<" ";

            }

            cout<<endl;

        }

        else

        {

            if(n<=26)

            {

                cout<<"-1"<<endl;

                continue;

            }

            map<int,int> mp;

            int x=3;

            for(int i=1;i<=n;i++)

            {

                if(i==1 || i==10 || i==26)

                    cout<<1<<" ";

                else if(i==23 || i==27)

                    cout<<2<<" ";

                else

                {

                    cout<<x<<" ";

                    mp[x]++;

                    if(mp[x]==2)

                    {

                        x++;

                    }

                }

            }

            cout<<endl;

        }

    }

 }

Codeforces数据结构：set\map\queue\stack

Codeforces第一题爱搞规律：奇数偶数，爱写while循环

### 原题58

[Problem - D - Codeforces](https://codeforces.com/contest/2037/problem/D)

思路：线性，一重循环。

主要学会如何使用数组排序，优先队列。

代码：

#include<bits/stdc++.h>

using namespace std;

#define int long long

signed main()

{

int t;

cin>>t;

while(t--)

{

int n,m,l;//n个障碍，m个能量装置

cin>>n>>m>>l;

//线性，要掌握队列（用不上先攒着）

vector< vector<int> > a;

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

{

int l,r;

cin>>l>>r;

a.push\_back({l,r,1});

}

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

{

int x,y;

cin>>x>>y;

a.push\_back({x,y,0});

}

sort(a.begin(),a.end());//给元素为数组的数组排序，优先排序l

priority\_queue<int> pq;

int cur=1,ans=0,flag=1;//目前弹跳力，答案

for(auto i:a)

{

if(i[2]==1)

{

int xu=i[1]-i[0]+2;//要避开的区间，连端点也要

while(cur<xu && pq.size())

{

auto j=pq.top();

pq.pop();

cur+=j;

ans++;

}

if(cur<xu)

{

flag=0;

break;

}

}

else if(i[2]==0)

{

pq.push(i[1]);

}

}

if(flag)

cout<<ans<<endl;

else

cout<<-1<<endl;

}

}

Codeforces构造题相当于规律题

### 原题59

[B-lz的数字问题\_牛客小白月赛105](https://ac.nowcoder.com/acm/contest/95937/B)

思路：善于用find(),substr()

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**signed** main()

{

    string a,b;

    cin>>a>>b;

**int** x=a.find('.');

**if**(x==-1)

        a=a+'.',x=a.size()-1;

**int** y=b.find('.');

**if**(y==-1)

        b=b+'.',y=b.size()-1;

**if**(a.substr(0,x)==b.substr(0,y))

    {

        string x1=a.substr(x+1,6);

        string y2=b.substr(y+1,6);

**while**(x1.size()<6)

        {

            x1 = x1+'0';

        }

**while**(y2.size()<6)

        {

            y2=y2+'0';

        }

**if**(x1==y2)

            cout<<"YES"<<endl;

**else**

            cout<<"NO"<<endl;

    }

**else**{

        cout<<"NO"<<endl;

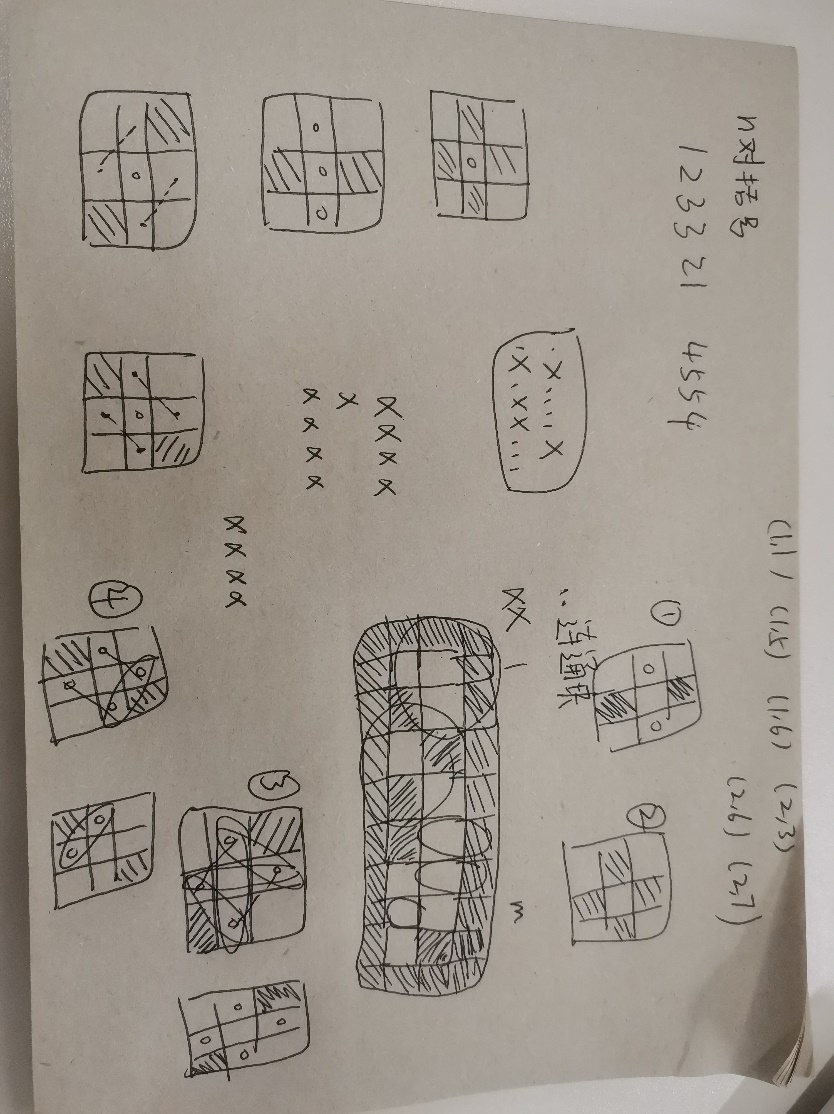
    }

}

### 原题60

[C-lz的蛋挞问题\_牛客小白月赛105](https://ac.nowcoder.com/acm/contest/95937/C)

思路：



周围围一圈奶贝，以每个蛋挞为核心的九宫格

代码：

#include<bits/stdc++.h>

**using** **namespace** std;

#define int long long

**signed** main()

{

    //输入，周围围一圈奶贝，以每个蛋挞为核心的九宫格

**int** m;

    cin>>m;

**char** ch[4][m+10];

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

    {

        cin>>ch[1][i];

    }

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

    {

        cin>>ch[2][i];

    }

**for**(**int** i=0;i<=m+1;i++)

    {

        ch[0][i]='x';

        ch[3][i]='x';

    }

**for**(**int** i=0;i<4;i++)

    {

        ch[i][0]='x';

        ch[i][m+1]='x';

    }

**int** ans=0;

**for**(**int** i=1;i<=2;i++)

    {

**for**(**int** j=1;j<=m;j++)

        {

**if**(ch[i][j]=='.')

            {

                //4大情况

**if**(ch[i-1][j]=='x'&&ch[i+1][j]=='x'&&ch[i][j-1]=='.'&&ch[i][j+1]=='.')

                {

                    ans++;

                }

**else** **if**(ch[i-1][j]=='x'&&ch[i+1][j]=='x'&&ch[i][j-1]=='x'&&ch[i][j+1]=='x')

                {

                    ans++;

                }

**else** **if**( ((ch[i-1][j]=='.'&&ch[i][j-1]=='.')||(ch[i][j+1]=='.'&&ch[i+1][j]=='.')) && (ch[i-1][j-1]=='x'&&ch[i+1][j+1]=='x')  )

                {

                    ans++;

                }

**else** **if**( ((ch[i-1][j]=='.'&&ch[i+1][j]=='.')||(ch[i][j-1]=='.'&&ch[i][j+1]=='.')) && (ch[i-1][j-1]=='x'&&ch[i+1][j+1]=='x')  )

                {

                    ans++;

                }

**else** **if**( ((ch[i-1][j]=='.'&&ch[i][j+1]=='.')||(ch[i][j-1]=='.'&&ch[i+1][j]=='.')) && (ch[i-1][j+1]=='x'&&ch[i+1][j-1]=='x')  )

                {

                    ans++;

                }

**else** **if**( ((ch[i-1][j]=='.'&&ch[i+1][j]=='.')||(ch[i][j-1]=='.'&&ch[i][j+1]=='.')) && (ch[i-1][j+1]=='x'&&ch[i+1][j-1]=='x')  )

                {

                    ans++;

                }

            }

        }

    }

    cout<<ans<<endl;

}

### 原题61

[Problem - B - Codeforces](https://codeforces.com/contest/2032/problem/B)

思路：构造题目，注意到数据范围，1<=bi<=n，让每个数成为众数

代码：

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  signed main()  {      int t;      cin>>t;      while(t--)      {          int n;          cin>>n;          int a[n];          stack<int> s;          map<int,int> mp;          for(int i=0;i<n;i++)          {              cin>>a[i];              mp[a[i]]++;          }          for(int i=1;i<=n;i++)          {              if(mp[i]==0)                  s.push(i);          }          set<int> c;          int b[n];          for(int i=0;i<n;i++)          {              if(c.size()!=0 && c.count(a[i]))              {                  b[i]=s.top();                  s.pop();              }              else              {                  c.insert(a[i]);                  b[i]=a[i];              }          }          for(int i=0;i<n;i++)              cout<<b[i]<<" ";          cout<<endl;      }  } |

### 原题62

[Problem - A - Codeforces](https://codeforces.com/contest/2049/problem/A)

思路：

回答只有0，1，2

0：全0

1：无0，00……，……00，00……00

2：剩下的

代码：

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  const int N=6000;  signed main()  {      int t;      cin>>t;      while(t--)      {          int n;          cin>>n;          int a[n];          map<int,int> mp;          vector<int> cnt;          for(int i=0;i<n;i++)          {              cin>>a[i];              mp[a[i]]++;              if(a[i]==0)                  cnt.push\_back(i);          }          if(mp[0]==0)          {              cout<<1<<endl;          }          else if(mp[0]==n)          {              cout<<0<<endl;          }          else          {              int flag=1;              if(cnt.size()==2 && cnt[0]==0&&cnt[1]==n-1 )              {                  flag=0;              }              if(a[0]==0 && a[n-1]==0)              {                  int ji=-1;                  for(int i=1;i<cnt.size();i++)                  {                      if(abs(cnt[i]-cnt[i-1])!=1)                      {                          ji=i;                          break;                      }                  }                  int f=1;                  for(int i=ji+1;i<cnt.size();i++)                  {                      if(abs(cnt[i]-cnt[i-1])!=1)                      {                          f=0;                          break;                      }                  }                  if(f)                      flag=0;              }              else if(a[0]==0)              {                  int f=1;                  for(int i=1;i<cnt.size();i++)                  {                      if(abs(cnt[i]-cnt[i-1])!=1)                      {                          f=0;                          break;                      }                  }                  if(f)                      flag=0;              }              else if(a[n-1]==0)              {                  int f=1;                  for(int i=cnt.size()-2;i>=0;i--)                  {                      if(abs(cnt[i+1]-cnt[i])!=1)                      {                          f=0;                          break;                      }                  }                  if(f)                      flag=0;              }                if(flag)                  cout<<2<<endl;              else                  cout<<1<<endl;          }      }  } |

### 原题63

[Problem - B - Codeforces](https://codeforces.com/contest/2049/problem/B)

思路：

Yes的条件很苛刻，只有

1.没有同时有s和p

2.同时有s和p

s……p，中间不能同时有s和p，不然不可以

……p，前面不能有p，不然不可以

s……,后面不能有s，不然不可以

代码：

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  const int N=6000;  signed main()  {      int t;      cin>>t;      while(t--)      {          int n;          cin>>n;          string s;          cin>>s;          map<char,int> mp;          for(int i=0;i<n;i++)          {              mp[s[i]]++;          }          if(mp['p']!=0 && mp['s']!=0)          {              if(s[0]=='s' && s[n-1]=='p')              {                  map<char,int> mp2;                  for(int i=1;i<n-1;i++)                  {                      mp2[s[i]]++;                  }                  if(mp2['p']!=0 && mp2['s']!=0)                  {                      cout<<"NO"<<endl;                  }                  else                      cout<<"YES"<<endl;              }              else if(s[n-1]=='p')              {                  map<char,int> mp3;                  for(int i=0;i<n-1;i++)                  {                      mp3[s[i]]++;                  }                  if(mp3['p']!=0)                  {                      cout<<"NO"<<endl;                  }                  else                      cout<<"YES"<<endl;              }              else if(s[0]=='s')              {                  map<char,int> mp4;                  for(int i=1;i<n;i++)                  {                      mp4[s[i]]++;                  }                  if(mp4['s']!=0)                  {                      cout<<"NO"<<endl;                  }                  else                      cout<<"YES"<<endl;              }              else                  cout<<"NO"<<endl;          }          else              cout<<"YES"<<endl;      }  } |

### 原题64

[Problem - E - Codeforces](https://codeforces.com/contest/2044/problem/E)

思路：二分

Y=x\*k^n,先定下k^n，只要枚举就可以，因为2^64就差不多超过1e18，那么只要枚举1~64就够了，然后再二分x的范围，找到第一个大于y范围的x，再找到最后一个小于y的范围的x，两者相减+1就是有序对的个数，注意特判一下，有没有超过x的范围！！！

代码：

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  int qpow(int a,int b)  {      int ret=1;      while(b)      {          if(b&1)              ret=ret\*a;          a=a\*a;          b>>=1;      }      return ret;  }  signed main()  {      int t;      cin>>t;      while(t--)      {          int k,l1,r1,l2,r2;          cin>>k>>l1>>r1>>l2>>r2;          int n;          for(int i=0;i<65;i++)          {              if((qpow(k,i))>r2)              {                  n=i;                  break;              }          }          int ans=0;          for(int i=0;i<n;i++)          {              int l=0,r=1000000007;//右边界l              while(l<r)              {                  int mid=l+r>>1;                  if(mid\*qpow(k,i)>r2)                      r=mid;                  else                      l=mid+1;              }              int l0=0,r0=1000000007;//左边界l0              while(l0<r0)              {                  int mid=l0+r0+1>>1;                  if(mid\*qpow(k,i)<l2)                      l0=mid;                  else                      r0=mid-1;              }              if(l-1<l1 || l0+1>r1)                  continue;              else                  ans += min(l-1,r1)-max(l1,l0+1)+1;          }          cout<<ans<<endl;      }  } |

### 原题65

[C-小红打怪\_牛客小白月赛104](https://ac.nowcoder.com/acm/contest/94879/C)

思路：二分答案，把答案当作已知

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **const** **int** N=200100;  **int** n,a[N],b[N];  **bool** check(**int** t)  {      //小红整体攻击  **for**(**int** i=0;i<n;i++)      {          b[i]=max(a[i]-t,0ll);      }      //队友2攻击  **int** c=t;  **for**(**int** i=1;i<n;i++)      {  **int** mi=min(min(b[i],b[i-1]),c);          c-=mi;          b[i]-=mi,b[i-1]-=mi;      }      c+=t;  **for**(**int** i=0;i<n;i++)      {          c-=b[i];      }  **return** c>=0;    }  **signed** main()  {      cin>>n;  **for**(**int** i=0;i<n;i++)      {          cin>>a[i];      }  **int** l=0,r=1000000007;  **while**(l<r)      {  **int** mid=(l+r)>>1;  **if**(check(mid))              r=mid;  **else**              l=mid+1;      }      cout<<l<<endl;  } |

### 原题66

[G-小红的数组操作(A组、B组)\_2024年第七届传智杯程序设计挑战赛初赛](https://ac.nowcoder.com/acm/contest/98215/G)

思路：二分答案，已知最大值是什么，那么循环跑一遍数组，凡是超过最大值的，我们就相应扣除操作次数，让这个数小于等于最大值，若操作次数<0，就不符合。

注意题目数据范围给错了，-1e18~1e9

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **const** **int** N=100100;  **int** n,k,x;  **int** a[N];  **bool** check(**int** t)  {  **int** c=k;  **for**(**int** i=0;i<n;i++)      {  **if**(a[i]<=t)  **continue**;          c-=(a[i]-t)/x;  **if**( (a[i]-t) %x )              c-=1;  **if**(c<0)  **return** **false**;      }  **return** **true**;  }  **signed** main()  {      cin>>n>>k>>x;  **for**(**int** i=0;i<n;i++)          cin>>a[i];  **int** l=-1e18,r=1e9;  **while**(l<r)      {  **int** mid=(l+r)>>1;  **if**(check(mid))              r=mid;  **else**              l=mid+1;      }      cout<<l<<endl;  } |

### 原题67

[Problem - B - Codeforces](https://codeforces.com/contest/2053/problem/B)

思路：前缀和，卡了边界悠着点

代码：

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  signed main()  {      int t;      cin>>t;      while(t--)      {          int n;          cin>>n;          string s="";          for(int i=0;i<n;i++)          {              s+='1';          }          set<int> st;          int l[n],r[n];          map<pair<int,int>,int> mp;          int m=0;          for(int i=0;i<n;i++)          {              cin>>l[i]>>r[i];              mp[{l[i],r[i]}]++;              if(l[i]==r[i])              {                  st.insert(l[i]);                  m=max(m,r[i]);              }          }          vector<int> s2;          int sum[m+10];          for(int i=0;i<=m;i++)          {              if(st.count(i))              {                  s2.push\_back(1);              }              else              {                  s2.push\_back(0);              }          }          sum[0]=s2[0];          for(int i=1;i<=m;i++)          {              sum[i]=sum[i-1]+(s2[i]);          }          for(int i=0;i<n;i++)          {              if(l[i]!=r[i] && l[i]<s2.size() && r[i]<s2.size() && sum[r[i]]-sum[l[i]-1]==r[i]-l[i]+1)              {                  s[i]='0';              }              if(mp[{l[i],r[i]}]>=2 && l[i]==r[i])              {                  s[i]='0';              }          }          cout<<s<<endl;      }    } |

### 原题68

[Problem - B - Codeforces](https://codeforces.com/gym/105491/problem/B)

思路：构造题目，最多k就是(n+1)/2\*（n+1）/2，超过不可以。构造回文便是，整个正方形左上角(n+1)/2\*（n+1）/2知道了，便可以知道整个，字母可以用周期（%k）

代码：

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  vector<char> v;  signed main()  {      for(char i='a';i<='z';i++)      {          v.push\_back(i);      }      int t;      cin>>t;      while(t--)      {          int n,k;          cin>>n>>k;          if(k<=((n+1)/2) \* ((n+1)/2) )          {              int x=0;              char ans[n][n];              for(int i=0;i<(n+1)/2;i++)              {                  for(int j=0;j<(n+1)/2;j++)                  {                      ans[i][j]=v[x];                      ans[i][n-j-1]=v[x];                      ans[n-i-1][j]=v[x];                      ans[n-i-1][n-j-1]=v[x];                      x = (x+1)%k;                  }              }              for(int i=0;i<n;i++)              {                  for(int j=0;j<n;j++)                  {                      cout<<ans[i][j];                  }                  cout<<endl;              }          }          else              cout<<"-1"<<endl;      }   } |

### 原题69

[数字组合](https://ac.nowcoder.com/acm/problem/235260)

思路：定一知一，写个双循环确定a+b组合有什么，c+d组合有什么，用map记录。枚举

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **signed** main()  {  **int** n;      cin>>n;      map<**int**,**int**> mp;  **int** a[n],b[n],c[n],d[n];  **for**(**int** i=0;i<n;i++)      {          cin>>a[i]>>b[i]>>c[i]>>d[i];      }  **for**(**int** i=0;i<n;i++)      {  **for**(**int** j=0;j<n;j++)          {              mp[c[i]+d[j]]++;          }      }  **int** ans=0;  **for**(**int** i=0;i<n;i++)      {  **for**(**int** j=0;j<n;j++)          {              ans += mp[-(a[i]+b[j])];          }      }      cout<<ans<<endl;  } |

### 原题70

[红色和紫色](https://ac.nowcoder.com/acm/problem/223888)

思路：博弈论，考虑n\*m为奇数，先手胜利

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **signed** main()  {  **int** n,m;      cin>>n>>m;      //n\*m为奇数，先手胜利  **if**(n&1 && m&1)          cout<<"akai"<<endl;  **else**          cout<<"yukari"<<endl;  } |

### 原题71

[Game](https://ac.nowcoder.com/acm/problem/201610)

思路：博弈论，算术基本定理，把数分解成质数

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **signed** main()  {  **int** n;      cin>>n;  **int** cnt=0;  **if**(n==1)      {          cout<<"Nancy"<<endl;  **return** 0;      }  **for**(**int** i=2;i<=n;i++)      {  **while**(n%i==0)          {              n /= i;              cnt++;          }      }  **if**(cnt&1)          cout<<"Nancy"<<endl;  **else**          cout<<"Johnson"<<endl;  } |

### 原题72

[取石子游戏 1](https://ac.nowcoder.com/acm/problem/50614)

思路：博弈论，先手有优势，只有n为(k+1)的倍数时，才可能后手赢

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **signed** main()  {  **int** n,k;      cin>>n>>k;  **if**(n%(k+1)==0)          cout<<2<<endl;  **else**          cout<<1<<endl;  } |

### 原题73

[石子游戏](https://ac.nowcoder.com/acm/problem/19156)

思路：博弈论，数操作数，奇数可分成1个偶数和1个奇数，那么操作数就是a[i]/2（分解操作数）+a[i]/2（产生的偶数）,再加上原来的偶数-1，就是合并操作数，全部加起来就是总的操作数，如果为奇数先手必胜

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **signed** main()  {  **int** n;      cin>>n;  **int** a[n];  **int** cnt1=0,cnt2=0;  **for**(**int** i=0;i<n;i++)      {          cin>>a[i];  **if**(a[i]&1)          {              cnt1 += a[i]/2+a[i]/2;          }  **else**              cnt2 ++;      }  **if**((cnt1+cnt2-1)%2==1)          cout<<"Alice"<<endl;  **else**          cout<<"Bob"<<endl;      } |

### 原题74

[[CQOI2009]中位数图](https://ac.nowcoder.com/acm/problem/19913)

思路：把小于k的数计为-1，把大于k的数计为1，求区间和为0的数量，让k当右侧，让k当左侧，包括k

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **signed** main()  {  **int** n,k;      cin>>n>>k;  **int** a[n+10];  **for**(**int** i=1;i<=n;i++)          cin>>a[i];  **int** index;  **for**(**int** i=1;i<=n;i++)      {  **if**(a[i]<k)              a[i]=-1;  **else** **if**(a[i]==k)              index=i,a[i]=0;  **else** **if**(a[i]>k)              a[i]=1;      }      map<**int**,**int**> mp;      map<**int**,**int**> mp2;      mp[index]=0;  **int** cnt=1;  **for**(**int** i=index+1;i<=n;i++)      {          mp[i]=a[i]+mp[i-1];          mp2[mp[i]]++;  **if**(mp[i]==0)              cnt++;      }  **for**(**int** i=index-1;i>=1;i--)      {          mp[i]=a[i]+mp[i+1];  **if**(mp[i]==0)              cnt++;          cnt += mp2[-mp[i]];      }      cout<<cnt<<endl;  } |

### 原题75

[C-猪猪养成计划1\_牛客小白月赛109](https://ac.nowcoder.com/acm/contest/99785/C)

思路：先把1~n只猪放在集合set里面，然后在里面二分找到大于等于l和小于等于r的数，每次找到的数记上顺序，然后从集合去掉。所以时间复杂度为线性

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **signed** main()  {  **int** n,q;      cin>>n>>q;      set<**int**> s;  **for**(**int** i=1;i<=n;i++)      {          s.insert(i);      }  **int** cnt=1;  **int** ans[n+10];  **memset**(ans,0,**sizeof** ans);  **while**(q--)      {  **int** op;          cin>>op;  **if**(op==1)          {  **int** l,r;              cin>>l>>r;  **while**(1)              {                  auto p=s.lower\_bound(l);  **if**(p==s.end() || \*p>r)  **break**;                  ans[\*p]=cnt;                  cnt++;                  s.erase(p);              }          }  **else** **if**(op==2)          {  **int** x;              cin>>x;              cout<<ans[x]<<endl;          }      }  } |

### 原题76

[矩阵消除游戏](https://ac.nowcoder.com/acm/problem/200190)

思路： 因为如果先贪心的选最大行或最大列，交叉选行和列，会有前对后的影响。所以我们直接状态压缩枚举每一行选不选，消除为0，再贪心选最大列  
1.如果k>n∣∣k>m*k*>*n*∣∣*k*>*m*，那么我们可以把矩阵全部拿完，此时令k=min(n,m)*k*=*min*(*n*,*m*)是不影响结果的，同时可以避免后面处理时把这种情况跳过了。  
2.枚举选那几行的情况通过二进制表示，标记那几行选了用01串表示，1表示选，0表示不选。  
3.确定了行后，计算每一列的和，剩下的次数选列的和最大的。

代码：

|  |
| --- |
| #include<bits/stdc++.h>  **using** **namespace** std;  #define int long long  **bool** cmp(**int** x,**int** y)  {  **return** x>y;  }  **signed** main()  {  **int** n,m,k;      cin>>n>>m>>k;  **int** a[n+10][m+10];  **for**(**int** i=0;i<n;i++)      {  **for**(**int** j=0;j<m;j++)          {              cin>>a[i][j];          }      }  **if**(k>n || k>m)          k=n;  **int** ans=0;  **for**(**int** i=0;i<(1<<n);i++)      {  **int** cnt=0;  **for**(**int** j=0;j<n;j++)          {  **if**(i>>j&1)                  cnt++;          }  **if**(cnt>k)  **continue**;  **int** cnt2=k-cnt;  **int** b[n+10][m+10];  **memcpy**(b,a,**sizeof** b);  **int** sum=0;  **for**(**int** j=0;j<n;j++)          {  **if**(i>>j&1)              {  **for**(**int** x=0;x<m;x++)                  {                      sum += b[j][x];                      b[j][x] = 0;                  }              }          }          vector<**int**> v;  **for**(**int** x=0;x<m;x++)                  {  **int** sum2=0;  **for**(**int** y=0;y<n;y++)                      {                              sum2 += b[y][x];                      }                      v.push\_back(sum2);                  }          sort(v.begin(),v.end(),greater<**int**>() );  **for**(**int** i=0;i<cnt2;i++)          {              sum += v[i];          }          ans=max(ans,sum);        }      cout<<ans<<endl;  } |

### 原题77

[Problem - D - Codeforces](https://codeforces.com/contest/2061/problem/D)

思路：逆向思维，我们思考什么样的两个数会合并成为我们想要的数

[x/2]向上取整，[x/2]向下取整，用优先队列存起来

代码：

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  signed main()  {      int t;      cin>>t;      while(t--)      {          int n,m;          cin>>n>>m;          int a[n],b[m];          int sum1=0,sum2=0;          map<int,int> mp;          priority\_queue<int> p,q;          for(int i=0;i<n;i++)          {              cin>>a[i];              sum1+=a[i];              p.push(a[i]);          }          for(int j=0;j<m;j++)          {              cin>>b[j];              sum2+=b[j];              q.push(b[j]);          }          if(sum1!=sum2 || n<m)          {              cout<<"No"<<endl;          }          else          {              while(1)              {                  if(p.empty() && q.empty())                  {                      break;                  }                  if(p.empty() || q.empty())                  {                      break;                  }                      int j=q.top();                      q.pop();                      int temp=p.top();                      if(j<temp)                      {                          break;                      }                      else if(j>temp)                      {                          q.push(j/2);                          q.push((j+1)/2);                      }                      else{                          p.pop();                      }                  }              if(p.empty() && q.empty())              {                  cout<<"Yes"<<endl;              }              else                  cout<<"No"<<endl;            }      }  } |

|  |
| --- |
| #include<bits/stdc++.h>  using namespace std;  #define int long long  signed main()  {      int t;      cin>>t;      while(t--)      {          int n,m;          cin>>n>>m;          int a[n],b[m];          int sum1=0,sum2=0;          map<int,int> mp;          priority\_queue<int> q;          for(int i=0;i<n;i++)          {              cin>>a[i];              sum1+=a[i];              mp[a[i]]++;          }          for(int j=0;j<m;j++)          {              cin>>b[j];              sum2+=b[j];              q.push(b[j]);          }          if(sum1!=sum2 || n<m)          {              cout<<"No"<<endl;          }          else          {              for(int i=0;i<3\*n+m;i++)              {                  if(!q.empty())                  {                      int j=q.top();                      q.pop();                      if(mp[j]!=0)                      {                          mp[j]--;                      }                      else                      {                          q.push(j/2);                          q.push((j+1)/2);                      }                  }                  else                  {                      break;                  }              }              if(q.empty())              {                  cout<<"Yes"<<endl;              }              else                  cout<<"No"<<endl;            }        }  } |

### 原题78

[Problem - B - Codeforces](https://codeforces.com/contest/2062/problem/B)

思路：贪心，来回扫一圈，看左右两边的步数是否超过本数