F&L

F题打表，遇见2^i>n则输出2^（i-1），唯一的难点是数据范围，java水过，代码：

import java.io.\*;

import java.util.\*;

import java.math.\*;

public class Main{

public static BigInteger two=BigInteger.valueOf(2);

public static BigInteger[] a=new BigInteger[1005];

public static void f(){

a[0]=BigInteger.ONE;

for(int i=1;i<=1000;i++)

a[i]=a[i-1].multiply(two);

}

public static void main(String[] args) {

int t;

Scanner cin=new Scanner(System.in);

t=cin.nextInt();

f();

while(t-->0){

BigInteger n=cin.nextBigInteger();

for(int i=1;i<=1000;i++){

if(a[i].compareTo(n)==1){

System.out.println(a[i-1]);

break;

}

}

}

}

}

L也是打表，但是递推关系需要找规律

代码：

import java.io.\*;

import java.util.\*;

import java.math.\*;

public class Main{

public static BigInteger two=BigInteger.valueOf(2);

public static BigInteger six=BigInteger.valueOf(6);

public static BigInteger[] a=new BigInteger[1005];

public static void f(){

a[1]=BigInteger.valueOf(3);

a[2]=BigInteger.valueOf(20);

for(int i=3;i<=1000;i++){

a[i]=a[i-1].multiply(six);

a[i]=a[i].subtract(a[i-2]).add(two);

//F题的表则为a[i]=a[i-1].multiply(two);

}

}

public static void main(String[] args){

int t;

Scanner cin=new Scanner(System.in);

t=cin.nextInt();

f();

while(t-->0){

BigInteger n=cin.nextBigInteger();

for(int i=1;i<=300;i++){

if(n.compareTo(a[i])==-1){

System.out.println(a[i]);

break;

}

//F题则n.compareTo(a[i])==-1，输出a[i-1]

}

}

}

}

J

所有的数其实可以分为3类，mod 3=0，或1，或2

0，0不能相邻，1，2不能相邻，用0隔开1和2，0，0之间需要错位，具体的看代码：

#include <bits/stdc++.h>

using namespace std;

int main(){

int t;

scanf("%d",&t);

while(t--){

int n,a[4]={0,0,0,0};

scanf("%d",&n);

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

int tmp;

scanf("%d",&tmp);

a[tmp%3]++;

}

if(a[0]<=n){

if(a[0]<=1&&a[1]>0&&a[2]>0) printf("NO\n");

else if(a[1]==0||a[2]==0) printf("YES\n");

else if(a[0]>=2&&a[1]>0&&a[2]>0){

if(a[0]==2&&a[1]%2==0&&a[2]%2==0) printf("NO\n");

else printf("YES\n");

}

}

else printf("NO\n");

}

}

I

这题好在数据小，暴力判断，但是有两个人在操作，一个想通过旋转选择的方阵最大化分数，一个想最小化，那可以在深搜的时候加一个标记，如果是A则深搜最大化分数，B则最小化

代码：

#include <bits/stdc++.h>

using namespace std;

int a[5][5],n,ans;

int dfs(int a[5][5],int cnt,int x,int y,int ans,int flag,int p1,int p2){

int sum=0,tmp[5][5];

for(int i=1;i<=4;i++)

for(int j=1;j<=4;j++) tmp[i][j]=a[i][j];

if(cnt){

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

for(int j=y;j<=y+1;j++) sum+=tmp[i][j];

int stmp=tmp[x][y];

tmp[x][y]=tmp[x][y+1];

tmp[x][y+1]=tmp[x+1][y+1];

tmp[x+1][y+1]=tmp[x+1][y];

tmp[x+1][y]=stmp;

}

if(cnt>=2\*n) return ans+sum;

for(int i=1;i<=3;i++){

for(int j=1;j<=3;j++){

if(!flag) p1=max(p1,dfs(tmp,cnt+1,i,j,ans+sum,!flag,p1,p2));

else p2=min(p2,dfs(tmp,cnt+1,i,j,ans+sum,!flag,p1,p2));

if(p1>=p2) i=4,j=4;

}

}

return flag?p2:p1;

}

int main(){

int t;

scanf("%d",&t);

while(t--){

scanf("%d",&n);

for(int i=1;i<=4;i++)

for(int j=1;j<=4;j++) scanf("%d",&a[i][j]);

printf("%d\n",dfs(a,0,0,0,0,0,-999999999,99999999));

}

}

H

暴力进化，具有耐力的选手写的这题，有问题请咨询他们（李嘉鑫，郑弘雨）

代码：

#include <stdio.h>

#include <string.h>

char map[655][655];

char tmp[655][655];

int moves[8][2] = {1, 0, -1, 0, 0, 1, 0, -1, 1, 1, 1, -1, -1, 1, -1, -1};

int heada, headb, a, b;

int main() {

int t;

scanf("%d", &t);

while (t --) {

memset(map, 0, sizeof(map));

int maxCnt = 0, maxG = 0;

char s[6];

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

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

scanf ("%s", s);

for (int j = 0; s[j]; j ++) {

if (s[j] == '#') {

map[325 + i][325 + j] = 1;

maxCnt ++;

}

}

}

heada = 325 - 1;

headb = 325 - 1;

a += 2;

b += 2;

int nextCnt;

for (int g = 1; g <= 321; g ++) {

nextCnt = nextG ();

if (nextCnt > maxCnt) {

maxCnt = nextCnt;

maxG = g;

}

}

printf("%d %d %d\n", maxG, maxCnt, nextCnt);

}

}

int judge(int a, int b) {

int cnt = 0;

for (int k = 0; k < 8; k++) {

int aa = a + moves[k][0];

int bb = b + moves[k][1];

if (map[aa][bb])

cnt++;

}

if (map[a][b]) {

if (cnt < 2 || cnt > 3) return 1;

} else {

if (cnt == 3) return 2;

}

return 4;

}

int nextG () {

int cnt = 0, maxi = 0, maxj = 0, mini = 999, minj = 999;

for (int i = heada; i < heada + a; i ++) {

for (int j = headb; j < headb + b; j ++) {

int res = judge(i, j);

if (res == 1) {

tmp[i][j] = 0;

}

else if (res == 2) {

tmp[i][j] = 1;

}

else {

tmp[i][j] = map[i][j];

}

if (tmp[i][j]) {

cnt ++;

if (i < mini) mini = i;

if (i > maxi) maxi = i;

if (j < minj) minj = j;

if (j > maxj) maxj = j;

}

}

}

for (int i = heada; i < heada + a; i++) {

for (int j = headb; j < headb + b; j++){

map[i][j] = tmp[i][j];

// if (tmp[i][j])

// putchar ('#');

// else

// putchar ('.');

}

//putchar ('\n');

}

heada = mini - 1;

headb = minj - 1;

a = maxi - mini + 3;

b = maxj - minj + 3;

//printf ("%d %d\n", heada, headb);

//printf ("%d %d %d %d\n", mini, minj, maxi, maxj);

return cnt;

}