(最长子回文)

#include<cstdio>

#include<cstring>

#include<cstdlib>

#include<cmath>

#include<algorithm>

#include<string>

using namespace std;

#define inf 1e-8

#define MAXN 100007

#define MAXNUM 200000000

typedef long long int64;

char s[MAXN];

int a[MAXN],fl[MAXN],fr[MAXN],sa[MAXN],myrank[MAXN],height[MAXN];

int wa[MAXN], wb[MAXN], wv[MAXN], wws[MAXN];

int rmq[50][MAXN];

int n,m,n1;

bool cmp(int \*wb, int a, int b, int l, int n){

int r,w;

r = a + l >= n ? 0 : wb[a + l];

w = b + l >= n ? 0 : wb[b + l];

return wb[a] == wb[b] && r == w;

}

void getsa(int \*a, int n, int m, int \*sa){

int i,j,k,r,w,p;

for(i=0; i<=m; i++) wws[i] = 0;

for(i=0; i<n; i++) wws[ wa[i] = a[i] ]++;

for(i=1; i<=m; i++) wws[i] += wws[i - 1];

for(i=n-1; i>=0; i--) sa[ --wws[ wa[i] ] ] = i;

for(j=1,p=1; j<n && p<n; j\*=2, m = p){

for(i=n-j,p=0; i<n; i++) wb[p++] = i;

for(i=0; i<n; i++) if(sa[i] >= j) wb[p++] = sa[i] - j;

for(i=0; i<=m; i++) wws[i] = 0;

for(i=0; i<n; i++) wv[i] = wa[ wb[i] ];

for(i=0; i<n; i++) wws[ wv[i] ]++;

for(i=1; i<=m; i++) wws[i] += wws[i-1];

for(i=n-1; i>=0; i--) sa[ --wws[ wv[i] ] ] = wb[i];

for(i=0; i<n; i++) wb[i] = wa[i];

for(i=1,p=1,wa[ sa[0] ] = 0; i<n; i++)

wa[ sa[i] ] = cmp(wb, sa[i], sa[i-1], j, n) ? p - 1 : p++;

}

}

void getheight(int \*a, int \*sa, int n, int \*height){

int i,j,k,r,w;

k = 0;

for(i=0; i<=n; i++) myrank[ sa[i] ] = i;

for(i=0; i<n; height[ myrank[i++] ] = k)

for(k ? k-- : 0, j = sa[ myrank[i] - 1]; a[i + k] == a[j + k]; k++);

}

void getrmq(int \*height, int n, int rmq[50][MAXN]){

int i,j,k,r,w,m;

m = (double)log((double)n + 1) / (double)log(2.0);

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

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

rmq[i][j] = MAXNUM;

for(i=0; i<=n; i++) rmq[0][i] = height[i];

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

for(j=0; j<=n - (1<<i) + 1; j++)

rmq[i][j] = min(rmq[i-1][j], rmq[i-1][j + (1 << (i - 1) ) ]);

}

int find(int rmq[50][MAXN], int l, int r){

int m = (double)log((double)r - l + 1) / (double)log(2.0);

return min(rmq[m][l], rmq[m][r - (1<<m) + 1]);

}

int lca(int rmq[50][MAXN], int l, int r){

int ll = myrank[l],rr = myrank[r];

if(ll > rr) swap(ll, rr);

ll++;

return find(rmq, ll, rr);

}

void init(){

int i,j,k;

n1 = n = strlen(s);

for(i=0; i<n; i++) a[i] = s[i];

for(i=n+1; i<=n+n; i++) a[i] = s[n + n - i];

a[n] = 200;

n = n + n + 1;

a[n] = 0;

getsa(a, n+1, 250, sa);

getheight(a, sa, n, height);

getrmq(height, n, rmq);

memset(fl, 10, sizeof(fl));

memset(fr, 0, sizeof(fr));

for(i=0; i<n1; i++){

k = lca(rmq, i, n - i - 1);

fl[i] = i - k + 1;

fr[i] = i + k - 1;

k = lca(rmq, i, n - i - 2);

fl[i] = min(fl[i], i - (k - 2));

fr[i] = max(fr[i], i + (k - 1));

//printf("%d : %d %d\n",i,fl[i],fr[i]);

}

}

int z[MAXN],top;

void solve(){

int i,j,k;

int ans = top = 0;

for(i=0; i<n1; i++){

if(top != 0 && fr[ z[top] ] >= fr[i] ) continue;

while(top != 0 && fl[i] <= fl[ z[top] ]) top--;

while(top > 1 && fl[i] <= fr[ z[top-1] ] + 1 ) top--;

z[++top] = i;

}

//for(i=1; i<=top; i++) printf("%d ",z[i]);

//printf("\n");

printf("%d\n",top-1);

}

int main(){

while(scanf("%s",s) != EOF){

init();

solve();

}

return 0;

}

高斯消元

#include<cstdio>

#include<cstring>

#include<algorithm>

#include<iostream>

using namespace std;

int map[40][40];

int ans[40];

int n;

void init()

{

int i,j,k,r,w;

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

memset(ans,0,sizeof(ans));

for(i=0;i<30;i++)

scanf("%d",&map[i][30]);

for(i=0;i<30;i++)

for(j=0;j<30;j++)

{ int x,y,xx,yy;

x = i/6; y = i % 6;

xx= j/6; yy= j % 6;

if(abs(x-xx) + abs(y-yy)<= 1 )

map[j][i] = 1;

}

//for(i=0;i<30;i++,printf("\n"))

//for(j=0;j<=30;j++)

//printf("%d",map[i][j]);

}

int gauss(int map[40][40],int ans[40])

{

int i,j,k,r,w;

for(k=0;k<30;k++)

{ i = k;

while(i<30 && map[i][k] == 0) i++;

if(i == 30) continue;

if(i > k)

{ for(j=0;j<=30;j++)

swap(map[i][j],map[k][j]);

}

for(i=0;i<30;i++)

if(map[i][k] && i != k)

{ for(j=k;j<=30;j++)

map[i][j] ^= map[k][j];

}

}

for(k=29;k>=0;k--)

{ ans[k] = map[k][30];

for(i=0;i<=30 && !map[k][i];i++) ;

if(i == 30) return 0;

for(i=k+1;i<30;i++)

ans[k] ^= map[k][i] \* ans[i];

//ans[k] ^= map[k][k];

}

return 1;

}

void solve()

{

int i,j,k,r,w;

if( !gauss(map,ans) )

printf("error\n");

for(i=0;i<30;i++)

{ printf("%d ",ans[i]);

if( (i+1) % 6 == 0) printf("\n");

}

}

int main()

{

int t,i;

//freopen("p1222.in","r",stdin);

scanf("%d",&t);

for(i=1;i<=t;i++)

{ init();

printf("PUZZLE #%d\n",i);

solve();

}

return 0;

}

**扫描线**

#include<cstdio>

#include<cstring>

#include<cstdlib>

#include<cmath>

#include<algorithm>

#include<string>

#include<vector>

using namespace std;

#define inf 1e-8

#define MAXN 2007

typedef long long int64;

int sgn(double x){

return x > inf ? 1: (x < -inf ? -1 : 0);

}

struct node{

double x,l,r;

int t;

node(double \_l, double \_r, double \_x,int \_t) : l(\_l), r(\_r), x(\_x), t(\_t) {}

bool operator < (const node &b) const {

return sgn(x- b.x) < 0;

}

};

vector<node> a;

int lazy[MAXN];

int cut[MAXN];

double fx[MAXN],fy[MAXN],sum[MAXN],num[MAXN],y[MAXN],ww[MAXN];

int n,m;

void init(){

int i,j,k,r,w;

double x1,y1,x2,y2;

double x[MAXN];

memset(lazy,0,sizeof(lazy));

m = 0;

a.clear();

// ax.clear();

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

scanf("%lf %lf %lf %lf",&x1,&y1,&x2,&y2);

a.push\_back( node(y1, y2, x1, 1) );

a.push\_back( node(y1, y2, x2, -1) );

//ax.push\_back( node(x1, x2, y1, 1) );

y[++m] = y2;

x[m] = x1;

y[++m] = y1;

x[m] = x2;

}

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

sort(y+1, y+m+1);

fy[1] = y[1];

w = 1;

for(i=2; i<=m; i++){

if(sgn(y[i] - y[i-1]) != 0)

fy[++w] = y[i];

}

memcpy(y, fy, sizeof(y));

m = w;

memset(fy,0,sizeof(fy));

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

fy[i] = fy[i-1] + y[i+1] - y[i];

memset(num, 0, sizeof(num));

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

num[i] = fy[i];

}

void updata(int t, int lc, int rc, int ll, int rr){

int mid = (ll+rr) >> 1;

if(cut[t] >= 1){

sum[lc] = num[mid] - num[ll-1];

sum[rc] = num[rr] - num[mid];

}

else{

sum[lc] = sum[rc] = 0;

}

cut[lc] = cut[rc] = cut[t];

lazy[lc] = lazy[rc] = 1;

lazy[t] = 0;

}

void getch(int t, int &lc, int &rc){

lc = t<<1;

rc = t<<1 | 1;

}

void add(int t, int ll, int rr, int l, int r, int h){

int lc,rc,mid;

if(rr < l || r < ll) return;

getch(t, lc, rc);

//printf("%d %d %d %d %.2f %.2f %d\n",l,r,ll,rr,num[ll-1],num[rr],sum[t]);

if(l <= ll && rr <= r){

cut[t] += h;

if(cut[t] >= 1){

sum[t] = num[rr] - num[ll-1];

}

else if(ll == rr) sum[t] = 0;

else sum[t] = sum[lc] + sum[rc];

//printf("%d %d %d %d %.2f %.2f %.2f %d\n",l,r,t,rr,sum[lc],sum[rc],sum[t],cut[t]);

return ;

}

mid = (ll + rr) >> 1;

add(lc, ll, mid, l, r, h);

add(rc, mid+1, rr, l, r, h);

if(cut[t] >= 1){

sum[t] = num[rr] - num[ll-1];

}

else sum[t] = sum[lc] + sum[rc];

//printf("== %f == %f == %f\n",sum[lc],sum[rc],sum[t]);

}

int find(double yy){

int l,r,mid;

l = 1; r = m;

while(l <= r){

mid = (l + r) / 2;

if(sgn(y[mid] - yy) > 0) r = mid - 1;

else if(sgn(y[mid] - yy) < 0) l = mid + 1;

else return mid;

}

return -1;

}

void solve(){

int i,j,k,r,l,w;

memset(cut,0,sizeof(cut));

memset(sum,0,sizeof(sum));

memset(lazy,0,sizeof(lazy));

memset(ww,0,sizeof(ww));

double ans = 0;

//printf("%d\n",m);

for(i=0; i<(int)a.size()-1; i++){

l = find(a[i].l);

r = find(a[i].r) - 1;

//printf("%f\n",fy[l]);

if(l <= r) add(1, 1, m-1, l, r, a[i].t);

//printf("%d %d %f %f %f\n",l, r, a[i].l, a[i].r,sum[1]);

ans += sum[1] \* (a[i+1].x - a[i].x);

}

printf("Total explored area: %0.2f\n",ans);

}

int main(){

int ca = 1,ok=0;

while(scanf("%d",&n) != EOF && n){

if(ok == 1) printf("\n");

init();

//printf("%d\n",m);

printf("Test case #%d\n",ca++);

solve();

ok = 1;

}

return 0;

}

**双强连通分量**

#include<cstdio>

#include<cstring>

#include<cstdlib>

#include<algorithm>

#define MAXN 1007

using namespace std;

int a[MAXN][MAXN],f[MAXN];

int n,m,ans;

void init(){

int i,j,k,r,w;

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

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

a[i][j] = a[j][i] = 1;

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

scanf("%d %d",&r,&w);

a[r][w] = a[w][r] = 0;

}

}

int zhan[MAXN],top,v[MAXN],df[MAXN],low[MAXN],num;

int d[MAXN];

bool pan(int w)

{

int i,j,k;

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

if(a[w][i] && v[i] != 0){

if(v[i] == 1){

v[i] = (v[w]-1)%2 + 2;

if( !pan(i) ) return false;

}

else if( (v[w]-1)%2 + 2 != v[i])

return false;

}

}

return true;

}

void dfs(int w,int fa){

int i,j,k,r;

df[w] = low[w] = ++num;

zhan[++top] = w;

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

if(a[w][i] && i != fa){

if(df[i] == 0){

dfs(i,w);

low[w] = min(low[w],low[i]);

if(low[i] >= df[w]){

memset(v,0,sizeof(v));

k = top;

do{

v[ zhan[top] ] = 1;

top--;

}while(zhan[top+1] != i);

v[w]=1;

if(!pan(w) ){

for(k=1;k<=n;k++)

if(v[k] >= 1) {

d[k] = 1;

}

}

}

}

else low[w] = min(low[w],df[i]);

}

}

void solve(){

int i,j,k,r,w;

ans = 0;

memset(f,0,sizeof(f));

top = num = 0;

memset(df,0,sizeof(df));

memset(low,0,sizeof(low));

memset(v,0,sizeof(v));

memset(d,0,sizeof(d));

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

if(df[i] == 0){

dfs(i,0);

}

}

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

if(d[i] == 0){

//printf("%d\n",i);

ans++;

}

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

}

int main()

{

while(scanf("%d %d",&n,&m) != EOF && n && m){

init();

solve();

}

return 0;

}

**半平面交角**

#include<cstdio>

#include<cstring>

#include<cstdlib>

#include<algorithm>

#include<cmath>

using namespace std;

#define MAXN 200

#define esp 1e-8

struct point{

double x,y;

point (double \_x=0,double \_y=0) : x(\_x) , y(\_y) {

}

void input(){

scanf("%lf %lf",&x,&y);

}

double len() const{

return sqrt(x\*x + y\*y);

}

};

struct line{

point s,e;

double ang;

};

int n;

line b[MAXN];

int sgn(double x){

return x > esp ? 1 : (x < -esp ? -1 : 0);

}

bool operator == (const point &p1,const point &p2){

return sgn(p1.x-p2.x) == 0 && sgn(p1.y-p2.y) == 0;

}

point operator - (const point &p1,const point &p2){

return point(p1.x-p2.x, p1.y-p2.y);

}

double operator \* (const point &p1,const point &p2){

return p1.x\*p2.y - p1.y\*p2.x;

}

bool para(line a,line b){

return sgn((a.e-a.s) \* (b.e-b.s)) == 0;

}

point JD(point a, point b, point c, point d){

double d1 = (b-a)\*(c-a),d2 = (b-a)\*(d-a),

d3 = (d-c)\*(a-c),d4 = (d-c)\*(b-c);

return point( (c.x\*d2 - d.x\*d1) / (d2-d1),

(c.y\*d2 - d.y\*d1) / (d2-d1));

}

point JD(line a, line b){

return JD(a.s, a.e, b.s, b.e);

}

void init()

{

int i,j,k;

point a[MAXN];

scanf("%d",&n);

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

a[i].input();

a[n] = a[0];

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

b[i].s = a[i+1];

b[i].e = a[i];

b[i].ang = atan2(a[i].x-a[i+1].x, a[i].y-a[i+1].y);

}

}

bool cmp(const line a, const line b){

if(sgn(a.ang - b.ang) != 0) return a.ang < b.ang;

else return sgn( (b.e - b.s) \* (a.e - b.s) ) > 0;

}

line d[MAXN];

//存的是逆时针的多边形

int solve()

{

int i,j,k,w;

int r,f,t;

t = 1;

sort(b,b+n,cmp);

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

if(sgn(b[i].ang - b[i-1].ang) != 0)

b[t++] = b[i];

if(t < 2) return 1;

r = 1; f = 0;

d[0] = b[0]; d[1] = b[1];

for(i=2;i<t;i++){

if( para(d[r],d[r-1]) || para(d[f+1],d[f]) ) return 0;

while(f < r && sgn( (b[i].e-b[i].s) \* (JD(d[r],d[r-1]) - b[i].s) )< 0 ) r--;

while(f < r && sgn( (b[i].e-b[i].s) \* (JD(d[f],d[f+1]) - b[i].s) )< 0 ) f++;

d[++r] = b[i];

}

while(f < r && sgn( (d[f].e-d[f].s) \* (JD(d[r],d[r-1]) - d[f].s) ) < 0) r--;

while(f < r && sgn( (d[r].e-d[r].s) \* (JD(d[f],d[f+1]) - d[r].s) ) < 0) f++;

return r > f+1;

}

int main()

{

int t,i;

scanf("%d",&t);

for(i=1;i<=t;i++){

init();

if(solve() ) printf("YES\n");

else printf("NO\n");

}

return 0;

}

Atan2(y,x)

返回给定的 X 及 Y 坐标值的反正切值。反正切的角度值等于 X 轴与通过原点和给定坐标点 (Y坐标, X坐标) 的直线之间的夹角。结果以弧度表示并介于 -pi 到 pi 之间（不包括 -pi）。

　　语法

　　ATAN2（Y坐标,X坐标）

　　X坐标 点的 X 坐标。

　　Y坐标 点的 Y 坐标。

　　说明

? 结果为正表示从 X 轴逆时针旋转的角度，结果为负表示从 X 轴顺时针旋转的角度。

KMP

void preKmp(char \*src, int m, int \*rule){  
  int i=0,j=-1;  
  rule[0]=j;  
  while(i<m){  
    while(j!=-1 && src[j] != src[i])  
      j=rule[j];  
    i++,j++;  
    if(j>=m)  
      rule[i]=rule[j-1];  
    else  
      rule[i]=j;  
  }  
}  
void kmp(char \*src, int m, char \*dest, int n){  
  int rule[MAXN];  
  preKmp(src,m,rule);  
  int i=0, j=0;  
  while(i<n){  
    while(j!=-1 && src[j] != dest[i]){  
      j = rule[j];  
    }  
    i++,j++;  
    if(j==m){  
      //print ans  
      j=rule[j];  
    }  
  }  
}

PRIME

int prime[664588], cnt = 0;

void makePrime() {

for (int i = 2; i < maxn; ++i) {

if (!f[i]) {

prime[cnt++] = i;

}

for (int j = 0; (int64)i \* prime[j] < maxn; ++j) {

f[i \* prime[j]] = true;

if (i % prime[j] == 0) {

break;

}

}

}

}