**Floyd+快速幂优化**

给你一个无向图，然后给了一个起点s和终点e，然后问从s到e的最短路是多少，中途有一个限制，那就是必须走k条边，路径可以反复走。

|  |
| --- |
| //STATUS:C++\_AC\_125MS\_1204KB  #include<stdio.h>  #include<stdlib.h>  #include<string.h>  #include<math.h>  #include<iostream>  #include<string>  #include<algorithm>  #include<vector>  #include<queue>  #include<stack>  using namespace std;  #define LL \_\_int64  #define pii pair<int,int>  #define Max(a,b) ((a)>(b)?(a):(b))  #define Min(a,b) ((a)<(b)?(a):(b))  #define mem(a,b) memset(a,b,sizeof(a))  #define lson l,mid,rt<<1  #define rson mid+1,r,rt<<1|1  const int N=210,M=1000000,INF=0x3f3f3f3f,MOD=1999997;  const LL LLNF=0x3f3f3f3f3f3f3f3fLL;  const double DNF=100000000;  int f[N\*10],w[N][N];  int nt,n,m,s,t;  struct Matrix{  int ma[N][N];  Matrix friend operator \* (const Matrix a,const Matrix b){  Matrix ret;  mem(ret.ma,0x3f);  int i,j,k;  for(k=0;k<n;k++)  for(i=0;i<n;i++)  for(j=0;j<n;j++)  if(a.ma[i][k]+b.ma[k][j]<ret.ma[i][j])  ret.ma[i][j]=a.ma[i][k]+b.ma[k][j];  return ret;  }  }mta,mtb;  int pow(int k)  {  mem(mtb.ma,0x3f);  for(int i=0;i<n;i++)mtb.ma[i][i]=0;  while(k){  if(k&1)mtb=mtb\*mta;  mta=mta\*mta;  k>>=1;  }  return mtb.ma[f[s]][f[t]];  }  int main()  {  // freopen("in.txt","r",stdin);  int i,a,b,c;  while(~scanf("%d%d%d%d",&nt,&m,&s,&t))  {  n=0;  mem(f,-1);  mem(mta.ma,0x3f);  for(i=0;i<m;i++){  scanf("%d%d%d",&c,&a,&b);  if(f[a]==-1)f[a]=n++;  if(f[b]==-1)f[b]=n++;  mta.ma[f[a]][f[b]]=mta.ma[f[b]][f[a]]=Min(mta.ma[f[a]][f[b]],c);  }  printf("%d\n",pow(nt));  }  return 0;  } |

SPFA栈实现

|  |
| --- |
| /\*  因为负数乘负数会变正，因此初始化d数组为0，起点设为1，从起点开始  更新各个点  \*/  #include <bits/stdc++.h>  using namespace std;  int n,m;  double road[1002][1002],d[1002];  using namespace std;  void init()  {  memset(d,0,sizeof(d));  }  void spfa(int from,int to)  {  memset(d,0,sizeof(d));  d[from]=1;  bool vis[1002];memset(vis,0,sizeof(vis));  vis[from]=1;  stack<int>P;  P.push(from);  while(!P.empty())  {  int v=P.top();  P.pop();  vis[v]=0;  for(int i=1;i<=n;i++)  {  if(v==i)continue;  if(d[i]<d[v]\*road[v][i])  {  d[i]=d[v]\*road[v][i];  if(!vis[i])  {  P.push(i);  vis[i]=1;  }  }  }  }  }  int main()  {  while(~scanf("%d",&n))  {  int i,j;  for(i=1;i<=n;i++)  {  for(j=1;j<=n;j++)  {  scanf("%lf",&road[i][j]);  }  }  int q;  scanf("%d",&q);  while(q--)  {  int x,y;  scanf("%d%d",&x,&y);  spfa(x,y);  if(d[y]==0)  printf("What a pity!\n");  else printf("%.3lf\n",d[y]);  }  }  return 0;  } |