struct node{

int id,w;

int next;

}side[Maxm];

int cnt=0,head[Maxn],vis[Maxn],dis[Maxn];

void init(){

memset(head,-1,sizeof(head));

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

memset(dis,0x3f3f3f3f,sizeof(dis));

cnt=0;

}

void add(int x,int y,int d){

side[cnt].id=y;

side[cnt].w=d;

side[cnt].next=head[x];

head[x]=cnt++;

}

struct Node{

int pos;

int cost;

Node(){}

Node(int pos,int cost):pos(pos),cost(cost){}

friend bool operator < (Node a,Node b){

return a.cost>b.cost;

}

};

void dist(int sx,int ex){

priority\_queue<Node>q;

q.push(Node(sx,0));

dis[sx]=0;

while(!q.empty()){

Node tmp=q.top();

q.pop();

if(vis[tmp.pos])continue;

vis[tmp.pos]=1;

if(tmp.pos==ex)break;

for(int i=head[tmp.pos];i!=-1;i=side[i].next){

int ans=side[i].id;

if(dis[ans]>dis[tmp.pos]+side[i].w){

dis[ans]=dis[tmp.pos]+side[i].w;

q.push(Node(ans,dis[ans]));

}

}

}

void Dijkstra(int s,int e)

{

dis[s] = 0;

sum[s]=sorce[s];

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

{

int minw = 0x3f3f3f3f,minv;

for(int i = 0;i< n;i++)//取最小权值点

{

if(vis[i]) continue;

if(dis[i]< minw)

minw = dis[i],minv = i;

}

vis[minv] = 1;//标记走过

if(minv == e) break;//遇终点break

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

{

if(vis[i]||!mp[minv][i]) continue;

if(dis[i]>dis[minv]+mp[minv][i]){

dis[i]=dis[minv]+mp[minv][i];

sum[i]=sum[minv]+sorce[i];

}

}

}