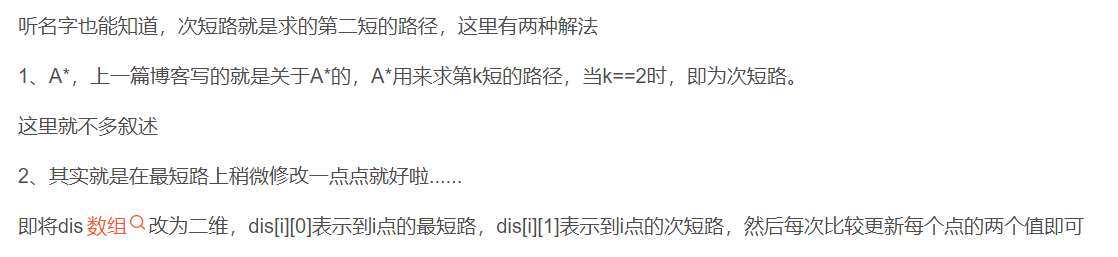


也就是求1到n的严格次短路



#include<bits/stdc++.h>

#define inf 1e18

#define met(a,x) memset((a),(x),sizeof((a)))

using namespace std;

const int N=1e5+10;

int n,m,s,t;

struct node

{

int v,c;

node(int \_v,int \_c):v(\_v),c(\_c){}

bool operator <(const node &p)const

{

return c>p.c;

}

};

struct edge

{

int v,cost;

edge(int \_v,int \_cost):v(\_v),cost(\_cost){}

};

vector<edge>e[N];

int dis[N][2];

void add(int u,int v,int w)

{

e[u].push\_back(edge(v,w));

}

void dj()

{

met(dis,inf);

dis[s][0]=0;

priority\_queue<node>q;

while(!q.empty())

q.pop();

q.push(node(s,0));

node tmp(0,0);

while(!q.empty()){

tmp=q.top();

q.pop();

int u=tmp.v;

if(dis[u][1]<tmp.c) //剪枝，跟最短路中vis数组作用一样

continue;

for(int i=0;i<e[u].size();i++){

int v=e[u][i].v;

int cost=e[u][i].cost;

int d=tmp.c+cost;

if(dis[v][0]>d){

swap(dis[v][0],d);

q.push(node(v,dis[v][0]));

}

if(dis[v][1]>d&&dis[v][0]<d){

dis[v][1]=d;

q.push(node(v,dis[v][1]));

}

}

}

}

int main()

{

while(cin>>n>>m)

{

int x,y,z;

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

e[i].clear();

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

scanf("%d%d%d",&x,&y,&z);

add(x,y,z);

add(y,x,z);

}

s=1,t=n;

dj();

cout<<dis[t][1]<<endl;

}

return 0;

}