

Educational Codeforces Round 102

[比赛链接](#)

E. Minimum Path

题意

给定 n 阶连通图，求点 1 到其他各点的最短路，其中路径长度的定义为所有边的权值和减去最大权值加上最小权值。

题解

不难发现所有边的权值和减去最大权值加上最小权值不大于所有边的权值和减去其中任意一条边权值再加上其中任意一条边权值。

不难发现将路径长度定义改为所有边的权值和减去其中任意一条边权值再加上其中任意一条边权值后，各点最短路值仍然不变。

于是设 $dis(i,0/1,0/1)$ 表示从点 1 到点 i 且是否减去一条边权值是否加上一条边权值的最小距离，直接跑最短路算法即可。

时间复杂度 $O(n \log m)$

```
const int MAXN=2e5+5;
int head[MAXN],edge_cnt;
struct Edge{
    int to,next,w;
}edge[MAXN<<1];
void Insert(int u,int v,int w){
    edge[++edge_cnt]=Edge{v,head[u],w};
    head[u]=edge_cnt;
}
struct Node{
    int u;
    LL dis;
    bool s1,s2;
    Node(int u=0,LL dis=0,bool s1=0,bool
s2=0):u(u),dis(dis),s1(s1),s2(s2){}
    bool operator < (const Node &b)const{
        return dis>b.dis;
    }
};
LL dis[MAXN][2][2];
bool vis[MAXN][2][2];
int main()
```

```
{
    int n=read_int(),m=read_int();
    mem(dis,-1);
    dis[1][0][0]=0;
    while(m--){
        int u=read_int(),v=read_int(),w=read_int();
        Insert(u,v,w);
        Insert(v,u,w);
    }
    priority_queue<Node>q;
    q.push(Node(1,0,0,0));
    while(!q.empty()){
        Node t=q.top();q.pop();
        if(vis[t.u][t.s1][t.s2])continue;
        vis[t.u][t.s1][t.s2]=true;
        for(int i=head[t.u];i;i=edge[i].next){
            int v=edge[i].to;
            _for(j,0,2)_for(k,0,2){
                LL d=t.dis+edge[i].w;
                bool s1=t.s1,s2=t.s2;
                if(j==1&&!s1){
                    d-=edge[i].w;
                    s1=true;
                }
                if(k==1&&!s2){
                    d+=edge[i].w;
                    s2=true;
                }
                if(dis[v][s1][s2]==-1||dis[v][s1][s2]>d){
                    dis[v][s1][s2]=d;
                    q.push(Node(v,d,s1,s2));
                }
            }
        }
    }
    _rep(i,2,n)
    space(dis[i][1][1]);
    return 0;
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string;jxm2001:contest:edu_102&rev=1610697649

Last update: 2021/01/15 16:00