**[**报告**]** **A - Invitation Cards**

**[Source]**

http://poj.org/problem?id=1511

**[Description]**

给出含n个节点有向图，求从点1到其他各点及其他各点到点1的最短路距离之和。即：∑d<1,i>+∑d<i,1>，其中1<i<=n。

**[Solution]**

用邻接表建图，同时建两个图，一个原图，一个反向图（所有边的方向和原图相反）。分别对原图和反向图以点1为源点求单源最短路（spfa），并把各最短路的值求和。注意求和后的数值可能超过32bit整数范围。

**[Code]**

#include<cstdio>  
#include<cstring>  
#include<queue>  
using namespace std;  
const int MAXN = 1000000+5, MAXM = 1000000+5;  
const long long INF = 0x3f3f3f3f3f3f3f3f;  
int T;  
int n, m, s, t, e;  
long long d[MAXN], w[2][MAXM];  
int v[2][MAXM], head[2][MAXN], next[2][MAXM];  
bool inq[MAXN];  
queue<int> Q;  
void addedge(int x, int y, int z)  
{  
    v[0][e] = y; w[0][e] = (long long)z;  
    next[0][e] = head[0][x]; head[0][x] = e;  
    v[1][e] = x; w[1][e] = (long long)z;  
    next[1][e] = head[1][y]; head[1][y] = e;  
    e++;  
}  
void SPFA(int p)  
{  
    for (int i = 1; i <= n; i++)  
        d[i] = (i == s ? 0 : INF);  
    memset(inq, 0, sizeof(inq));  
    Q.push(s); inq[s] = 1;  
    while (!Q.empty())  
    {  
        int u = Q.front(); Q.pop();  
        inq[u] = 0;  
        for (int e = head[p][u]; e != -1; e = next[p][e])  
            if (d[v[p][e]] > d[u]+w[p][e])  
            {  
                d[v[p][e]] = d[u]+w[p][e];  
                if(!inq[v[p][e]])  
                {  
                    Q.push(v[p][e]);  
                    inq[v[p][e]] = 1;  
                }  
            }  
    }  
}  
int main()  
{  
    scanf("%d", &T);  
    while (T--)  
    {  
        scanf("%d%d", &n, &m);  
        e = 0;  
        memset(head, -1, sizeof(head));  
        for (int i = 1; i <= m; i++)  
        {  
            int x, y, z;  
            scanf("%d%d%d", &x, &y, &z);  
            addedge(x, y, z);  
        }  
        s = 1;  
        long long ans = 0;  
        SPFA(0);  
        for (int i = 2; i <= n; i++)  
            ans += d[i];  
        SPFA(1);  
        for (int i = 2; i <= n; i++)  
            ans += d[i];  
        printf("%I64d\n", ans);  
    }  
    return 0;  
}