K短路

```
/*
 * Dijkstra变形，可以证明每个点经过的次数为小于等于K，
 * 所有Dijkstra的数组dist由一维变为二维，记录经过该点
 * 1次、2次......k次的最小值
 * 输出dist[n - 1][k]即可
 */
#include<bits/stdc++.h>
using namespace std;
int g[1010][1010];
int n, m, x;
const int INF = 0x3f3f3f3f;
int vis[1010];
int dist[1010][20];

int main(int argc, const char * argv[])
{
    while (cin >> n >> m >> x)
    {
        //初始化
        memset(g, 0x3f, sizeof(g));
        memset(dist, 0x3f, sizeof(dist));
        memset(vis, 0, sizeof(vis));
        for (int i = 0; i < m; i++)
        {
            int p, q, r;
            cin >> p >> q >> r;
            if (r < g[p][q])
            {
                g[p][q] = r;
            }
        }
        dist[1][0] = 0;
        dist[0][0] = INF;

        while (1)
        {
            int k = 0;
            for (int i = 1; i <= n; i++)
            {
                if (vis[i] < x && dist[i][vis[i]] < dist[k][0])
                {
                    k = i;
                }
            }
            if (k == 0)
            {
                break;
            }
            if (k == n && vis[n] == x - 1)
            {
                break;
```

```
            }
            for (int i = 1; i <= n; i++)
            {
                if (vis[i] < x && dist[k][vis[k]] + g[k][i] < dist[i][x])
                {
                    dist[i][x] = dist[k][vis[k]] + g[k][i];
                    for (int j = x; j > 0; j--)
                    {
                        if (dist[i][j] < dist[i][j - 1])
                        {
                            swap(dist[i][j], dist[i][j - 1]);
                        }
                    }
                }
            }
            vis[k]++;
        }

        if (dist[n][x - 1] < INF)
        {
            cout << dist[n][x - 1] << endl;
        }
        else
        {
            cout << -1 << endl;
        }
    }
    return 0;
}
```