*int* n, m;

    std::cin >> n >> m;

    std::vector<std::vector<ll>> vec(n + 1, std::vector<ll>(n + 1, INT\_MAX));

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

    {

        ll x, y, l;

        std::cin >> x >> y >> l;

*if*(x != y) *//去掉重边和自环*

            vec[x][y] = std::min(vec[x][y], l);

*//因为要使用边权所以使用邻接矩阵来存图*

    }

    std::vector<ll> dis(n + 1, INT\_MAX);

    dis[1] = 0;

    std::set<*int*> se;

    std::function<ll()> dijsitra = [&]() -> *ll{*

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

        {

*int* t = -1;

*for*(*int* j = 1; j <= n; j++)

            {

*if*(!se.count(j))

*if*(t == -1 || dis[t] > dis[j])

                        t = j;

            }

*for*(*int* j = 1; j <= n; j++)

            {

                dis[j] = std::min(dis[j], dis[t] + vec[t][j]);

            }

            se.insert(t);

        }

*return* (dis[n] == INT\_MAX ? -1 : dis[n]);

    };

注意dijstra只能处理没有负权边的情况，因为边带权，所以我们使用邻接矩阵来存图，同时排除掉自环与重边的影响。Se表示已经确认了最短路的点，每次找出dis最小的点t来确认最短路并更新其他路径的最短路。