## Prim算法

适于稠密图。

从点出发，先加入一个点，然后不断寻找未加入的点中，与已加入的点距离最近的点，将它加入。

实际实现时当然不可能每次枚举每一个未求解点与所有已求解点的距离，类比Dijkstra算法，开一个lowC数组存每一个未求解点到当前树的最短距离，起初只加入0号节点，那么lowC[i]=cost[0][i]，每次从lowC数组中找最小值，对应节点就是新求解的点。每次有求出新结点后，假设编号为x，更新一下lowC，即看lowC[i]改成cost[x][i]是否会更小，会就替换。

邻接矩阵写法占空间，无法表示多重边，但很容易定位到指定边。

## Kruskal算法

适于稀疏图。

先将所有的边按从小到大排序，然后从小到大遍历，只要加入当前边不会形成环就加入，判断环可使用并查集。