

HW7

林宸昊 PB20000034

1.
 - 删去的这条边不在原有最小生成树中，则原有最小生成树即为新图最小生成树；
 - 删去的这条边在原有最小生成树中，原有最小生成树被分割为两个集合，则遍历所有剩下的边，找到连通这两个集合的最短横切边。
 - 总时间代价为 $O(E)$ ，查找删去边的两个顶点集合仅需对最小生成树中边进行遍历。
2.
 - 首先找到原生成树中所有与新增边的两点相连的所有边（时间代价 $O(V)$ ）并删去；
 - 然后重新寻找两个集合中权重最小的横切边加入到生成树中。
 - 总时间代价为 $O(V)$
3.
 - 设定一个新单源点，该源点到所有原起点的距离为0，由于没有负边，仅采取一次Dijkstra算法即可；
 - 为了能够得到所给任意起点到达每个顶点的最短路径，对于所给七点都可以维护一个一维数组，根据最短路径树每个结点均仅有一个父节点这一特点，从最后终点逐步向上层遍历即可得到路径，如果某顶点不在该数组中则返回NULL即可。
4. 对S的每个结点运行Dijkstra算法，直到出现下列情况（其中每次都将开始节点标记为已访问）：
 - 到达T中的某个节点；
 - 到达S中某个已被访问的节点；
 - 已没有可访问的节点。

对于Dijkstra算法而言，worst case为 $O(E \lg V)$ ，但对于以上修改使得该算法并不需要运行 $|S|$ 次，针对所访问的边来说，其实仅完整运行了一次Dijkstra算法，也就是说最坏情况下所需时间应为 $O(E \lg V)$ 。