

3 C

先考虑所有边权都不为0的情况应该怎么做。

首先不难看出，点集在树上的斯坦纳树即为其虚树。虚树上的边权和即为正确答案。

我们下面证明：该算法是正确的，当且仅当给定点集 V_1 在 G 上的虚树点集 V'_1 和 V_1 相同。（另一个说法是，任意 V_1 中的两个点 u, v 的LCA也在 V_1 中）

其充分性不难证明，下面证其必要性：

1

若存在属于虚树点集 V'_1 ，且不属于 V_1 的点（下面简称虚点），则在虚树上，虚点 u 至少存在三条相关联的边。在斯坦纳树中，这三条边对答案的贡献是这三条边的权值和；而在我们建立的最小生成树中，至少有一条边的权值被计算了两次（由于虚点并不在原先的点集中，所以要让这三条边的另一个端点联通，就不得不经其中一条边两次），因而答案就是错误的。

对于边权存在0的情况，只需要将所有由0边连结的连通块视为一个点，一个连通块被选中当且仅当其中至少有一个点被选中，然后应用上述做法即可。

关于如何维护虚点：可以考虑倒着做，每次删去一个点。如果删去的这个点有三条及以上的边相连，它就转型成为虚点。如果某次删除后，某个虚点只有两条边相连，则该虚点消失。答案为1当且仅当不存在虚点时。