# Test About Graph - solution

## 1 calc

### 1.1 solution

按 Kruskal 算法依次考虑最小生成树中的每条边。对于一条权重为 $weight$ 的边 $(x, y)$ ，当将 $x$ 所在连通块与 $y$ 所在连通块合并时，实际上这两个连通块之间的所有边的权重都不会小于 $weight$ ，即最小权重为 $weight$ 。

也就是说，我们按权重从小到大枚举最小生成树中的每一条边 $(x, y)$ ，每次将答案加上 $weight_{x,y} \times Size_x \times Size_y$ 即可。

### 1.2 complexity

时间复杂度 $O(N \log N)$ 。

空间复杂度 $O(N)$ 。

## 2 unique

### 2.1 solution

考虑 Kruskal 算法。有这样一个结论：

对于任意 $P$ ，当将所有权重 $\leq P$ 的边加入后并运行算法后，图的连通性是不变的。

也就是说，对于权重相同的边，我们将它们一并处理，重点是判断这些边的选择方案是否唯一。这个问题很好解决，实际上我们只需要判断这些边是否形成了"环"即可。注意，这里的"环"是指在将连通块看做点的情况下的环。

### 2.2 complexity

时间复杂度 $O(TM \log M)$ 。

空间复杂度 $O(M)$ 。

# 3 patrol

## 3.1 solution

关键是如何剔除经过了一条边两次的那些"最优答案"。容易知道，这条边肯定是和 1 点直接相连（否则最优答案不会重复经过这条边)。考虑任意一个点 $k$ ，我们可以得到从 1 点到该点的最短路径，并不妨假设这条路径经过的第一条边为 $E$ 。但是，有可能通过该点拓展出一个不合法的"最优答案"。注意到从 $k$ 点到 1 点的最短路径是确定的，也就是说最后经过的边肯定为 $E$ 。我们只需要找一条从 1 点到 $k$ 点的次短路径，使得经过的第一条边不为 $E$ 即可。

最终，我们只需要对每个点记录两条路径即可：一条是最短路径，另一条是经过的第一条边不与最短路径经过的第一条边相同的次短路径。

实现的时候以 $SPFA$ 为佳。

# 4 flow

## 4.1 solution

纯粹的网络流。

注意要将一个点 $x$ 拆成两个 $s_x, t_x$ ，然后连一条容量为 $Limit_x$ 的边 $(s_x, t_x)$ 。这样可以限制点的容量。