

# Kruskal算法

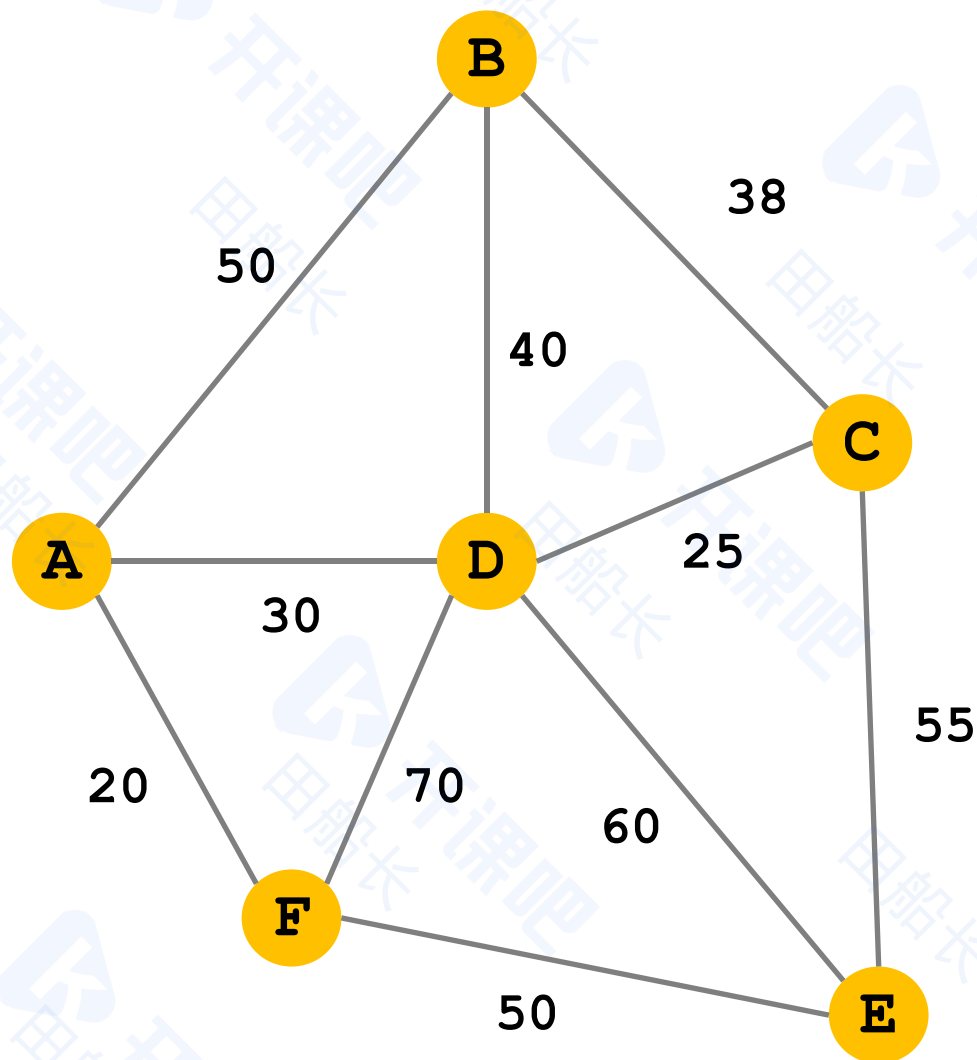
田船长

# Kruskal 算法

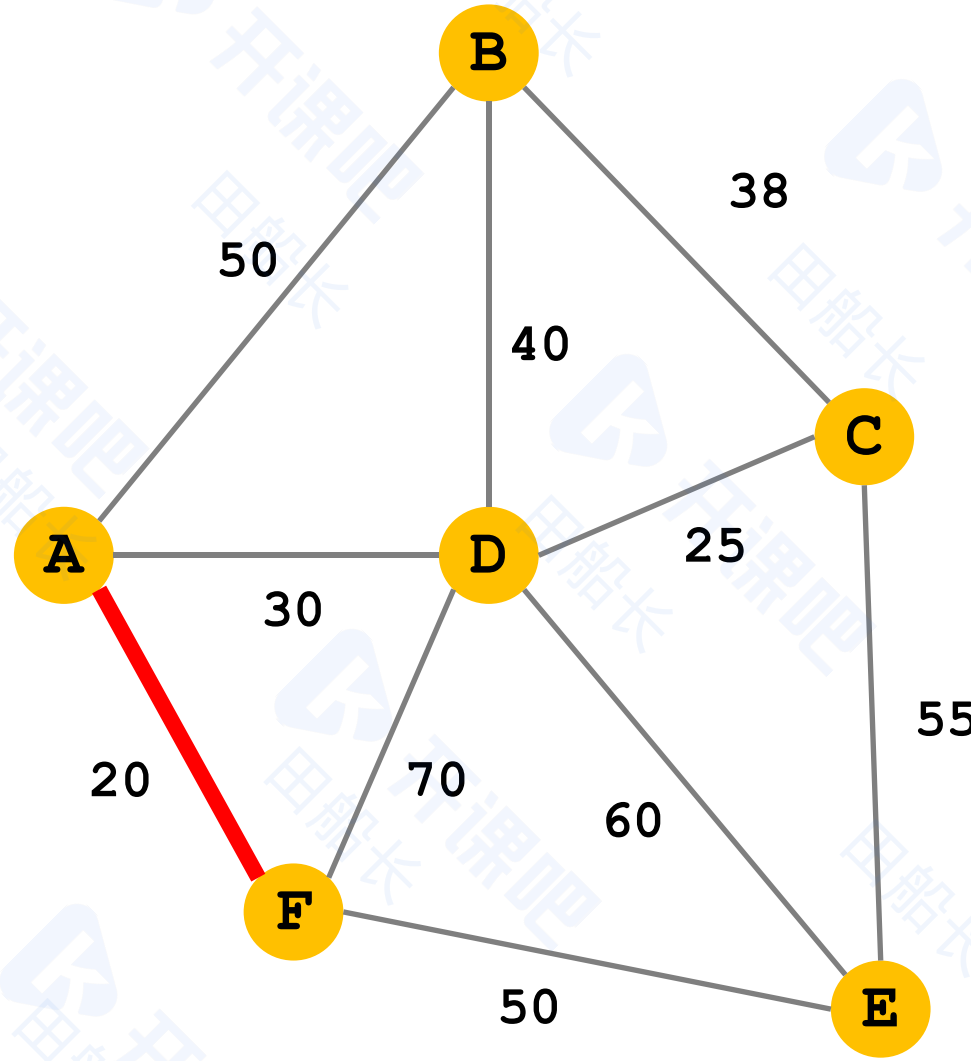
1. 初始化并查集，将所有边按照权值排序
2. 按照权值从小到大，依次遍历所有边
3. 遍历一条边时，判断这条边的两个端点是否相连（并查集）
4. 若相连，则这条边不是最小生成树中的一条边
5. 若不相连，则这条边是最小生成树中的一条边，连通这两点
6. 继续这个过程，直到所有点均已连通

# Kruskal 算法

使用Kruskal算法  
求解最小生成树

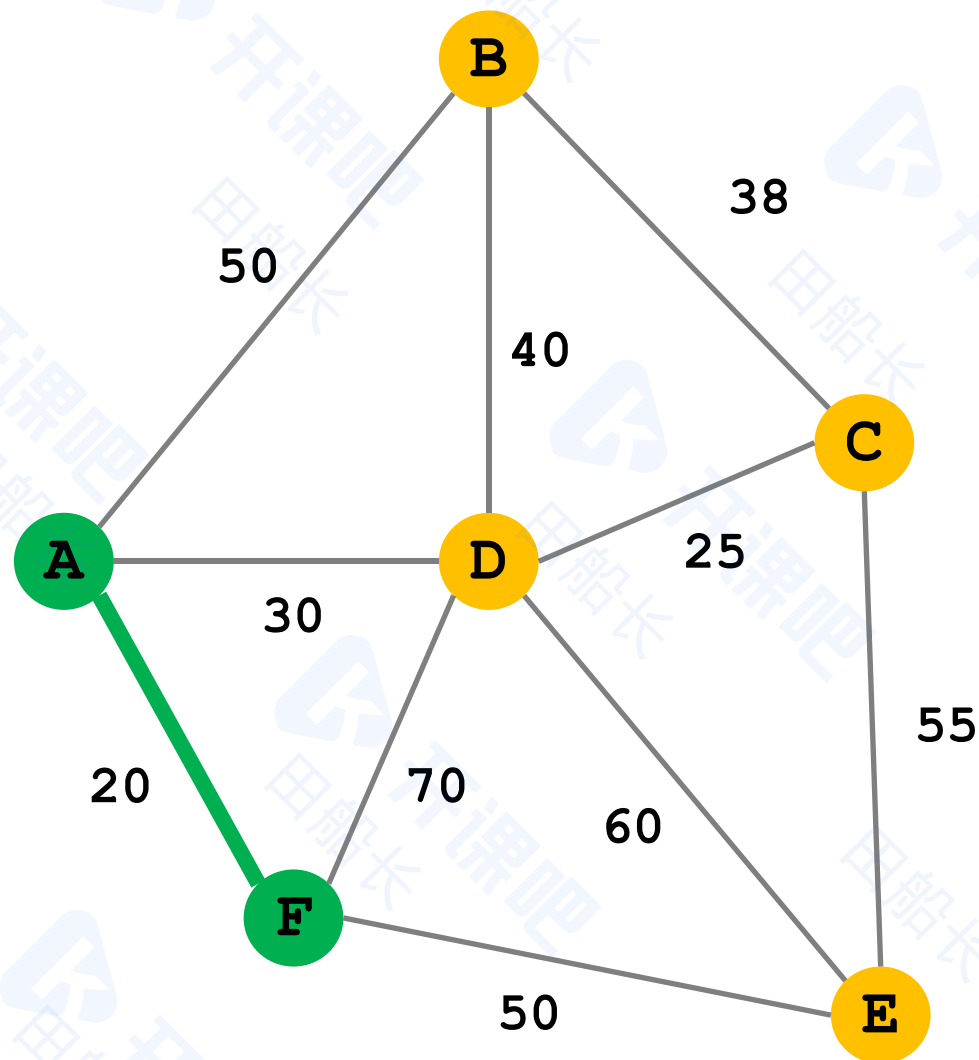


# Kruskal 算法



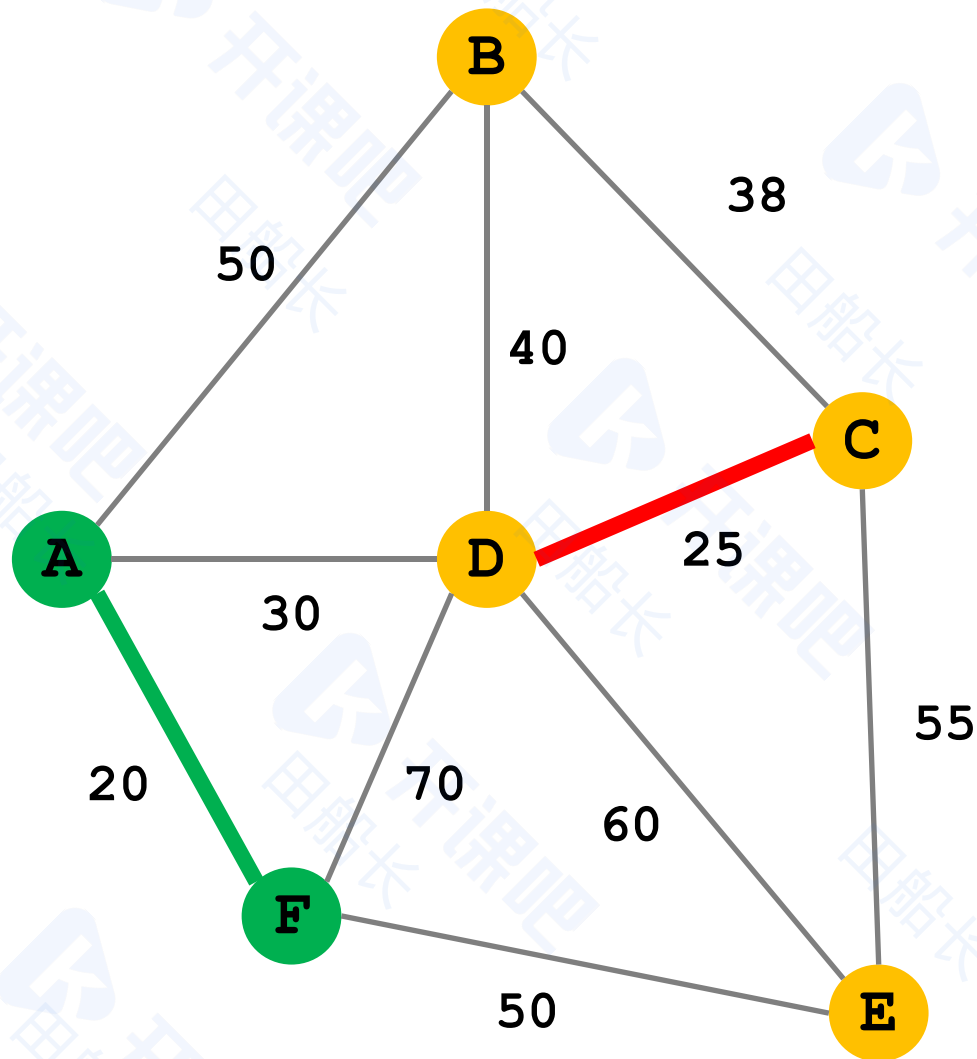
按照权值从小到大  
依次遍历所有边  
当前遍历的边用红色表示

# Kruskal 算法



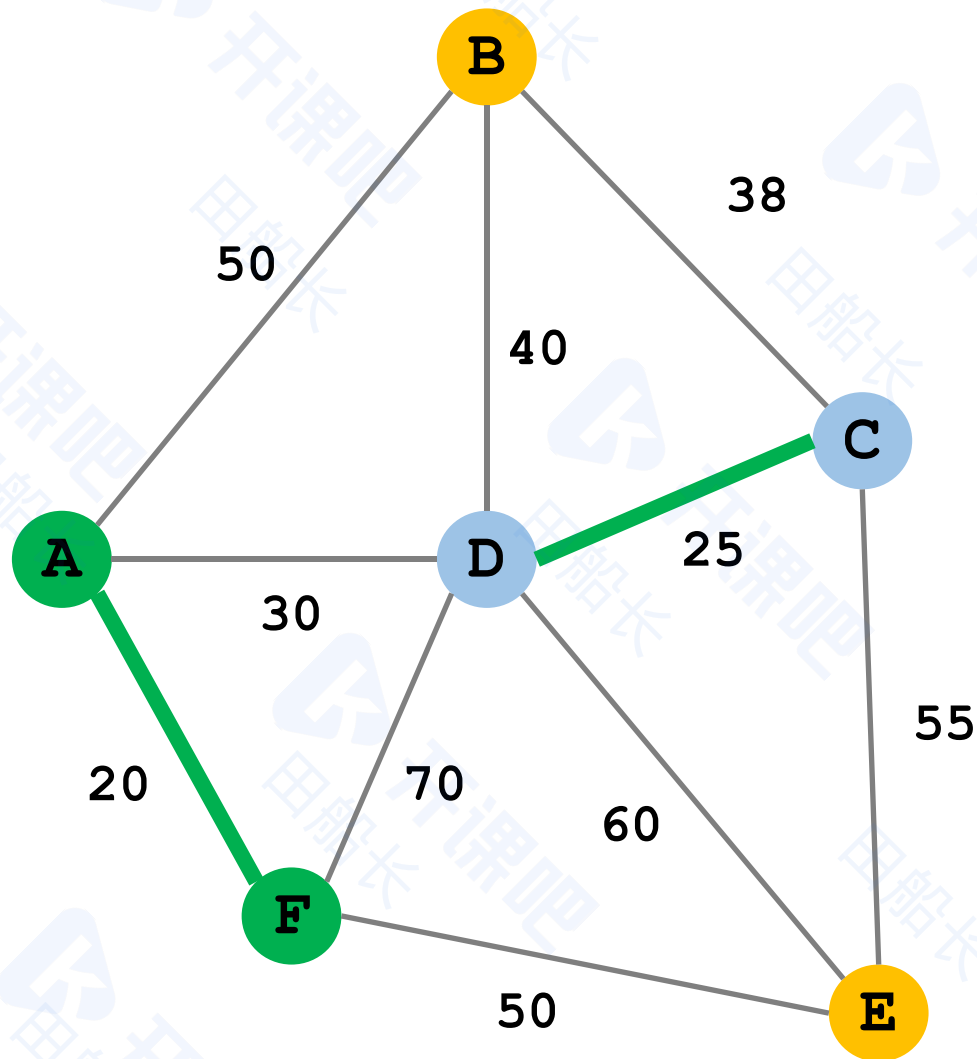
这条边的两个端点不相连  
选中这条边  
连通这两点

# Kruskal 算法



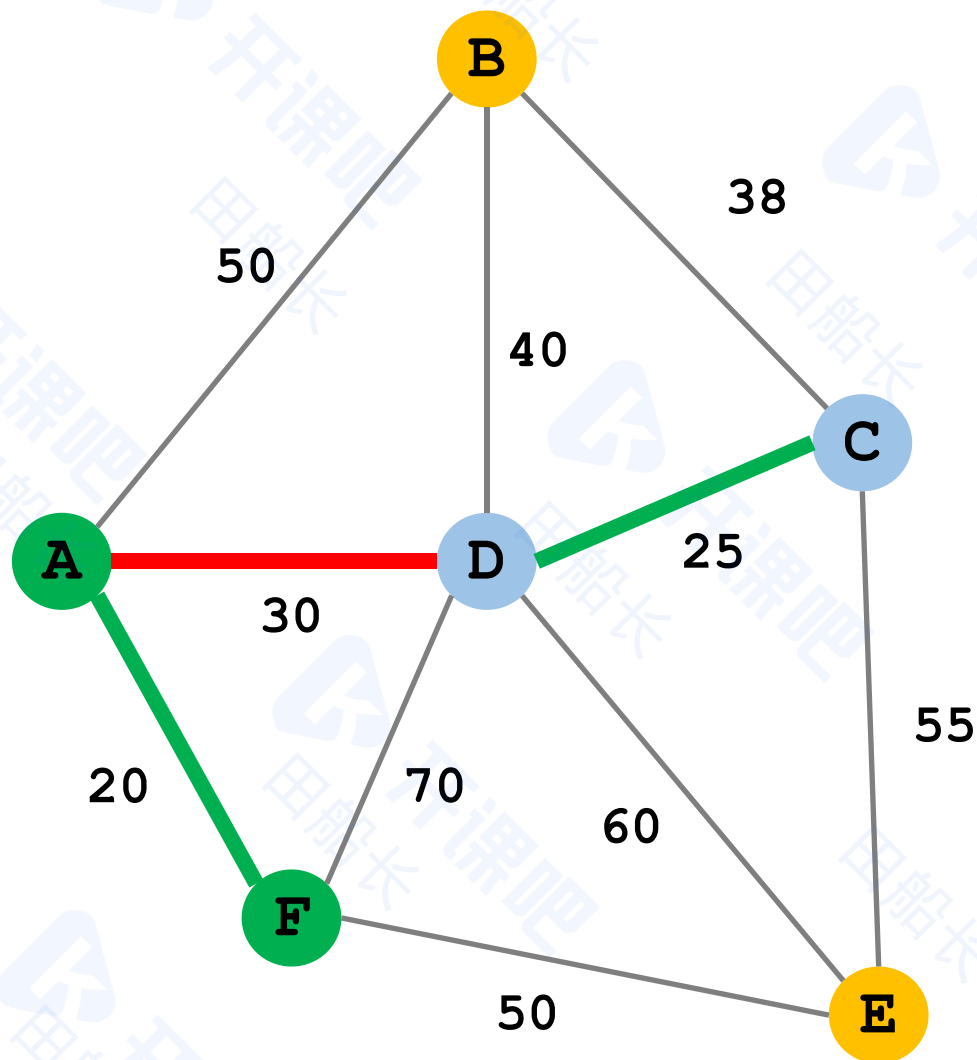
按照权值从小到大  
依次遍历所有边  
当前遍历的边用红色表示

# Kruskal 算法



这条边的两个端点不相连  
选中这条边  
连通这两点

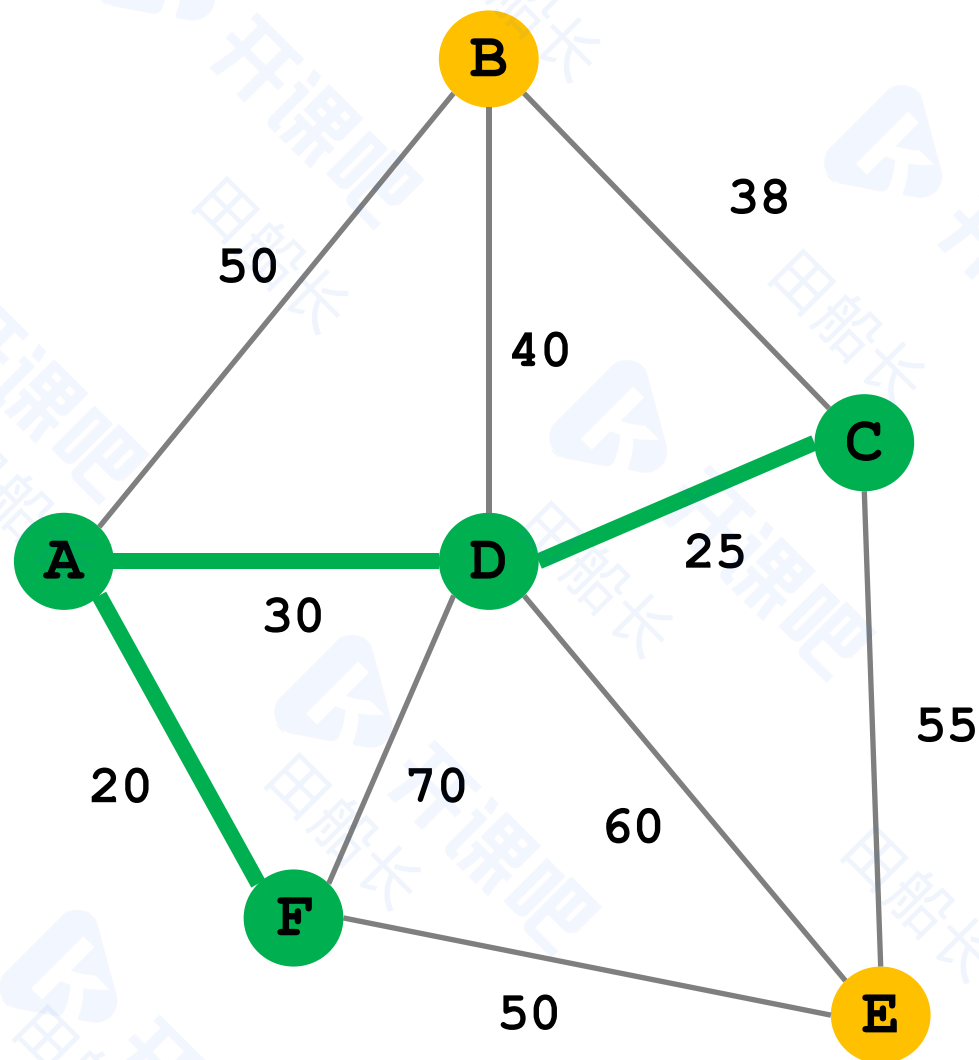
# Kruskal 算法



按照权值从小到大  
依次遍历所有边  
当前遍历的边用红色表示

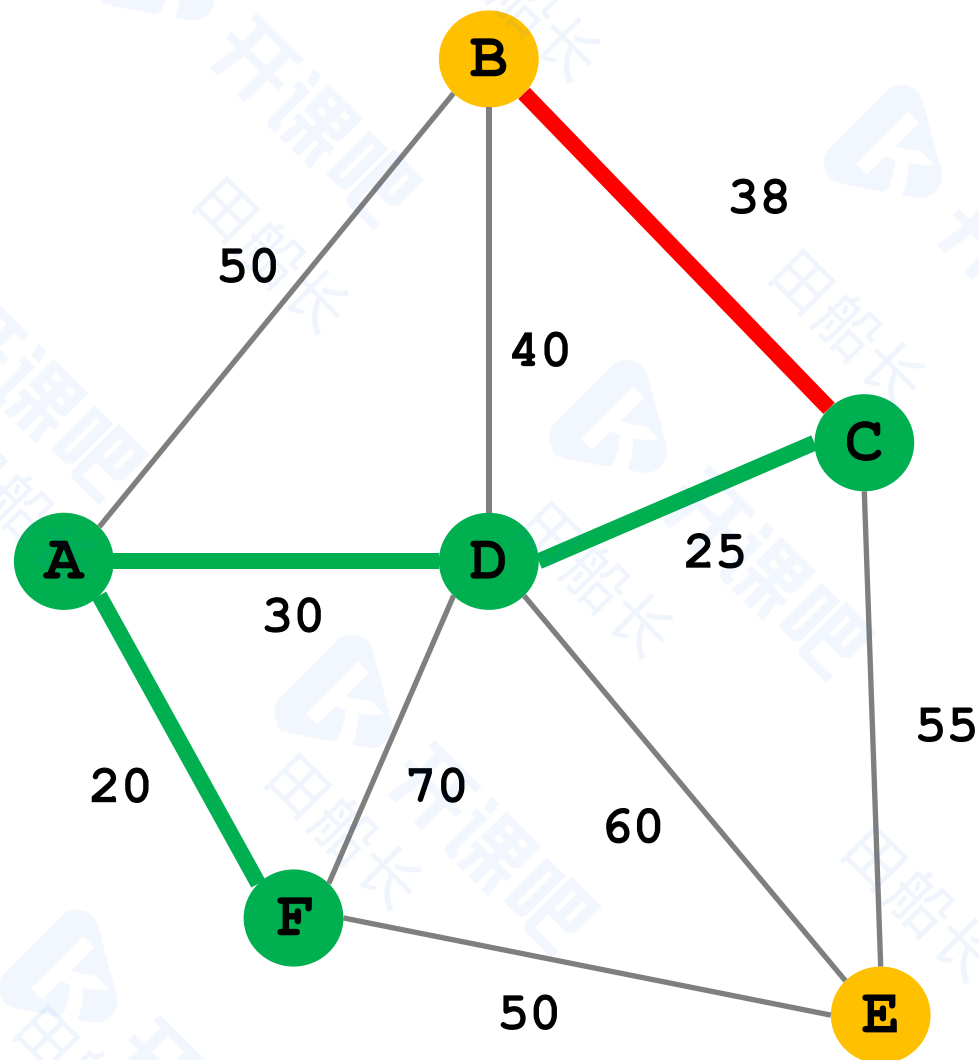


# Kruskal 算法



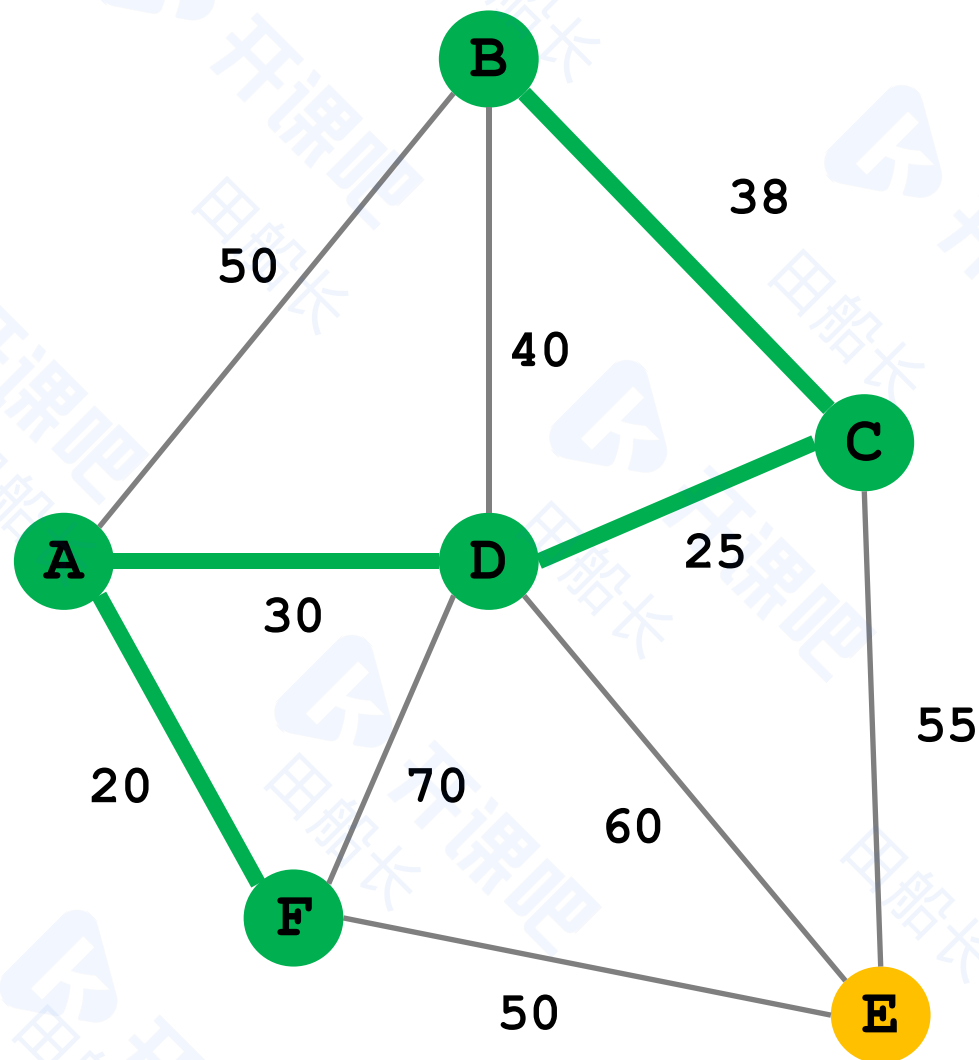
这条边的两个端点不相连  
选中这条边  
连通这两点

# Kruskal 算法



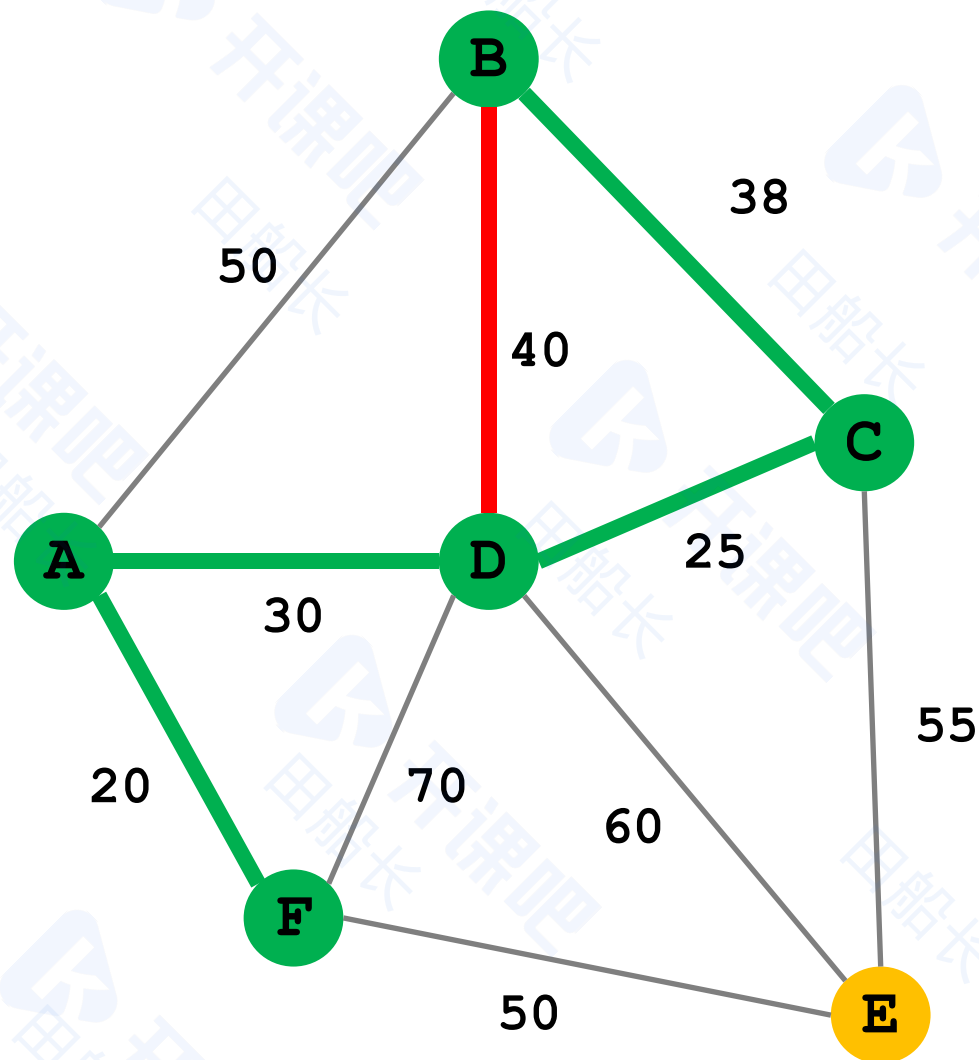
按照权值从小到大  
依次遍历所有边  
当前遍历的边用红色表示

# Kruskal 算法



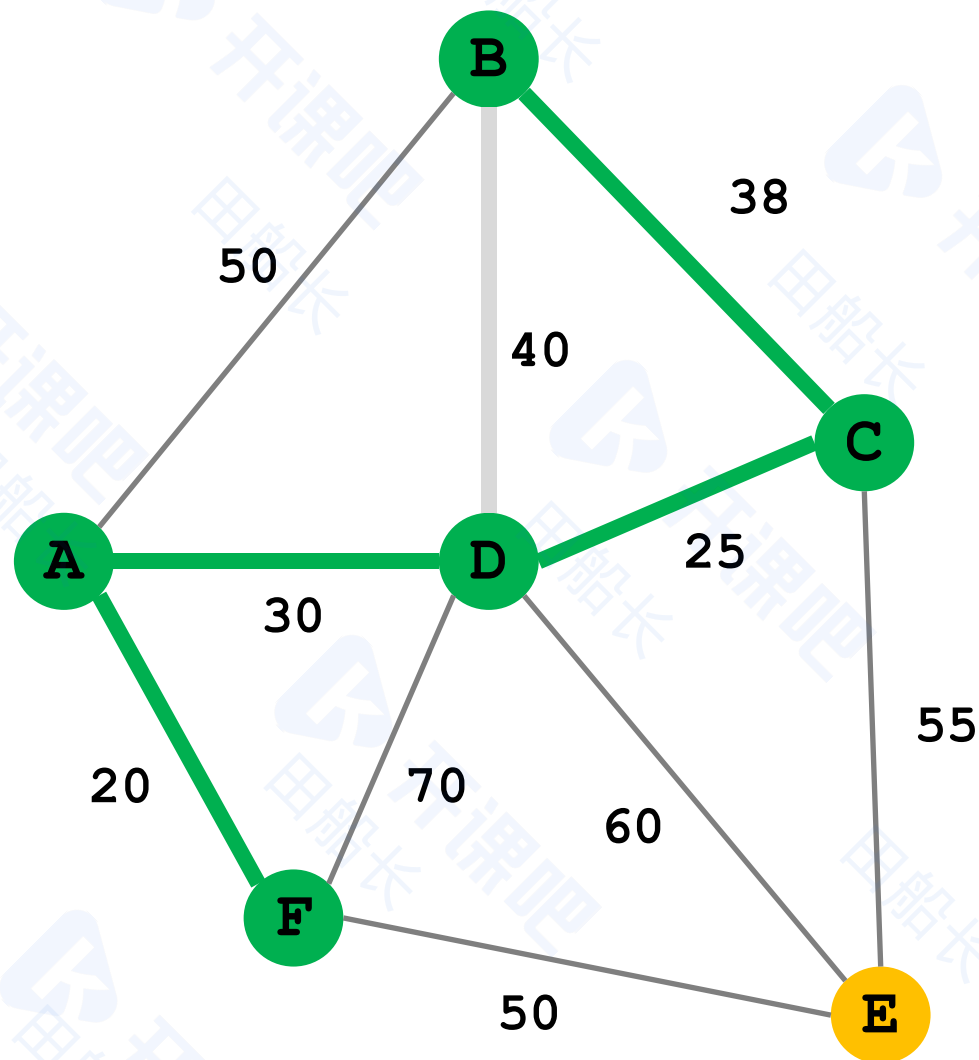
这条边的两个端点不相连  
选中这条边  
连通这两点

# Kruskal 算法



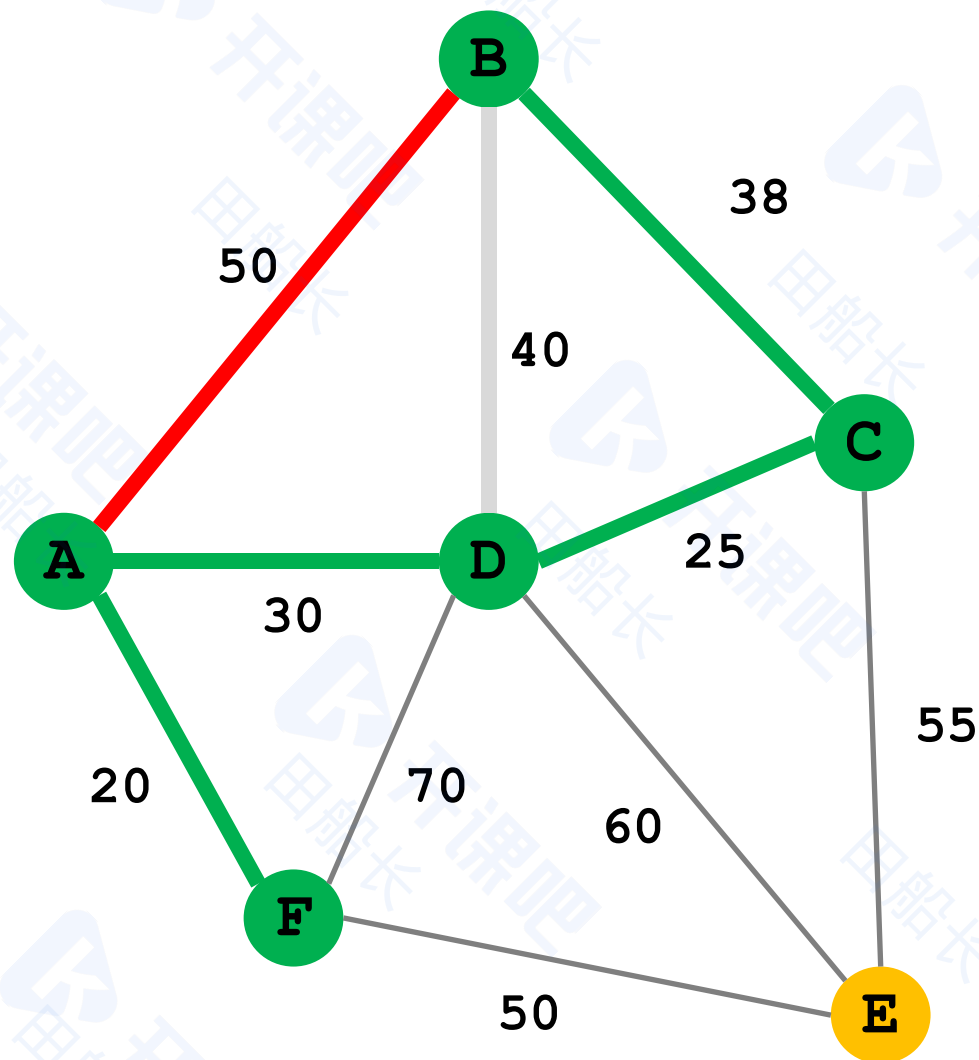
按照权值从小到大  
依次遍历所有边  
当前遍历的边用红色表示

# Kruskal 算法



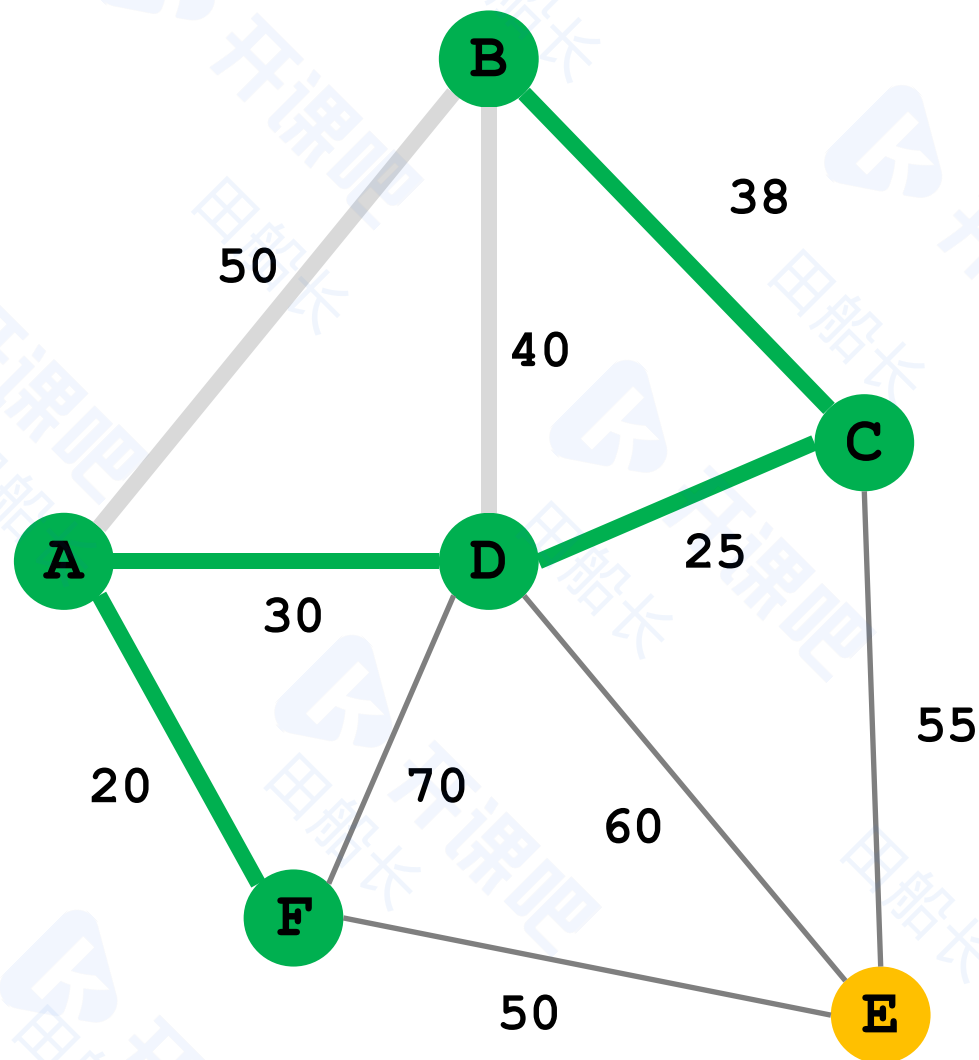
这条边的两个端点相连  
什么也不做

# Kruskal 算法



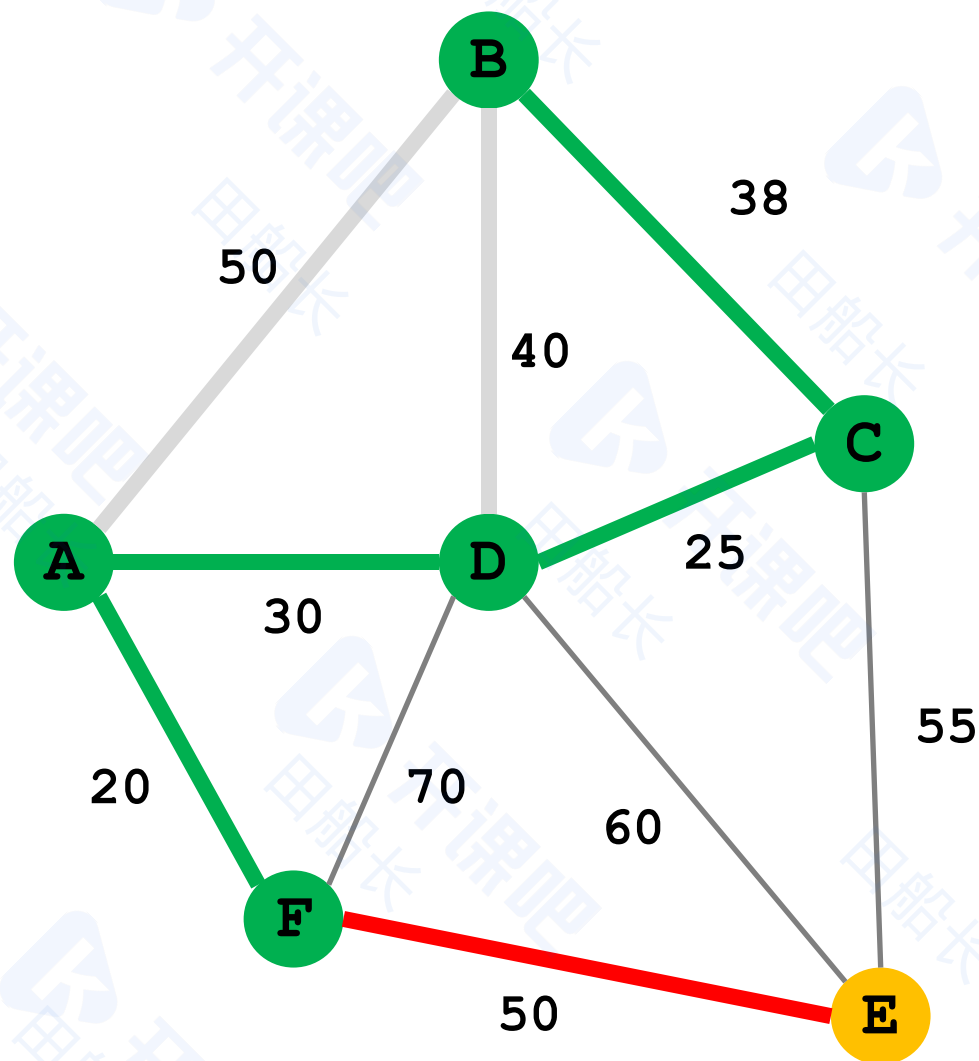
按照权值从小到大  
依次遍历所有边  
当前遍历的边用红色表示

# Kruskal 算法



这条边的两个端点相连  
什么也不做

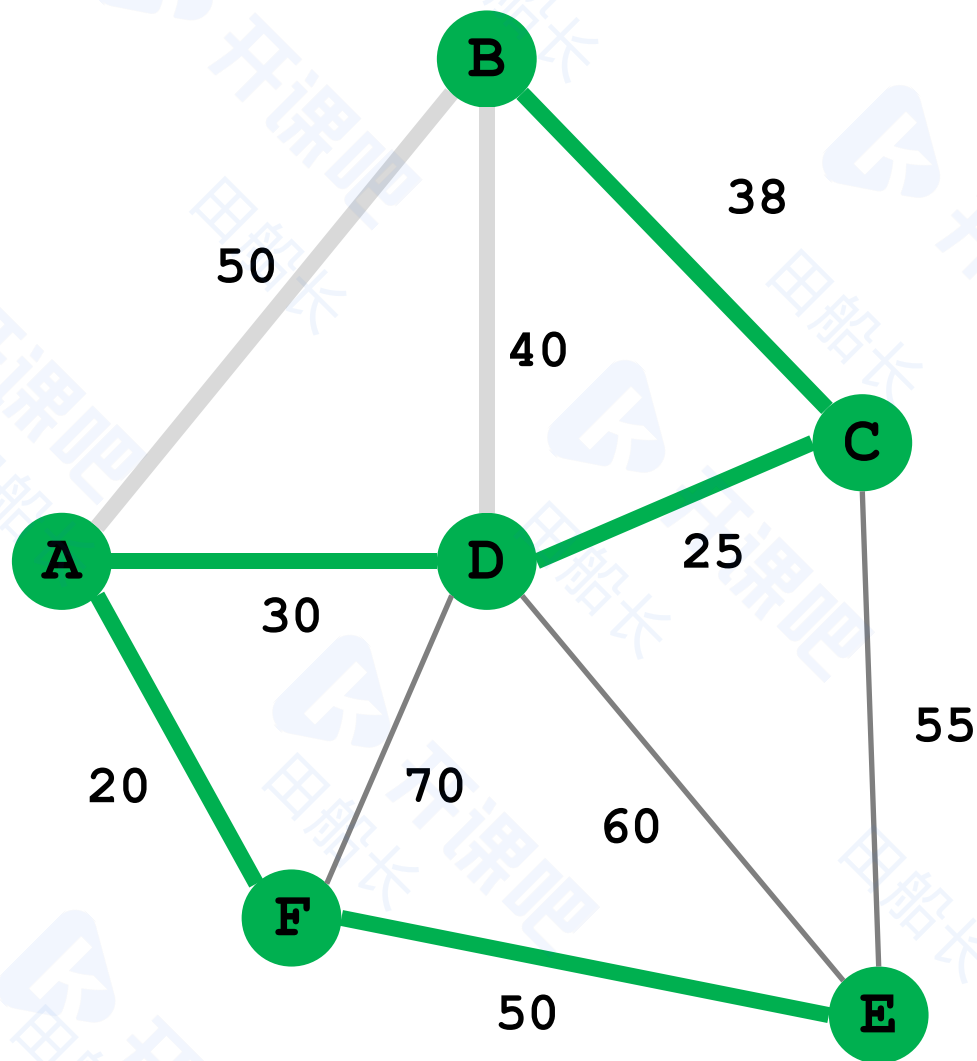
# Kruskal 算法



按照权值从小到大  
依次遍历所有边  
当前遍历的边用红色表示

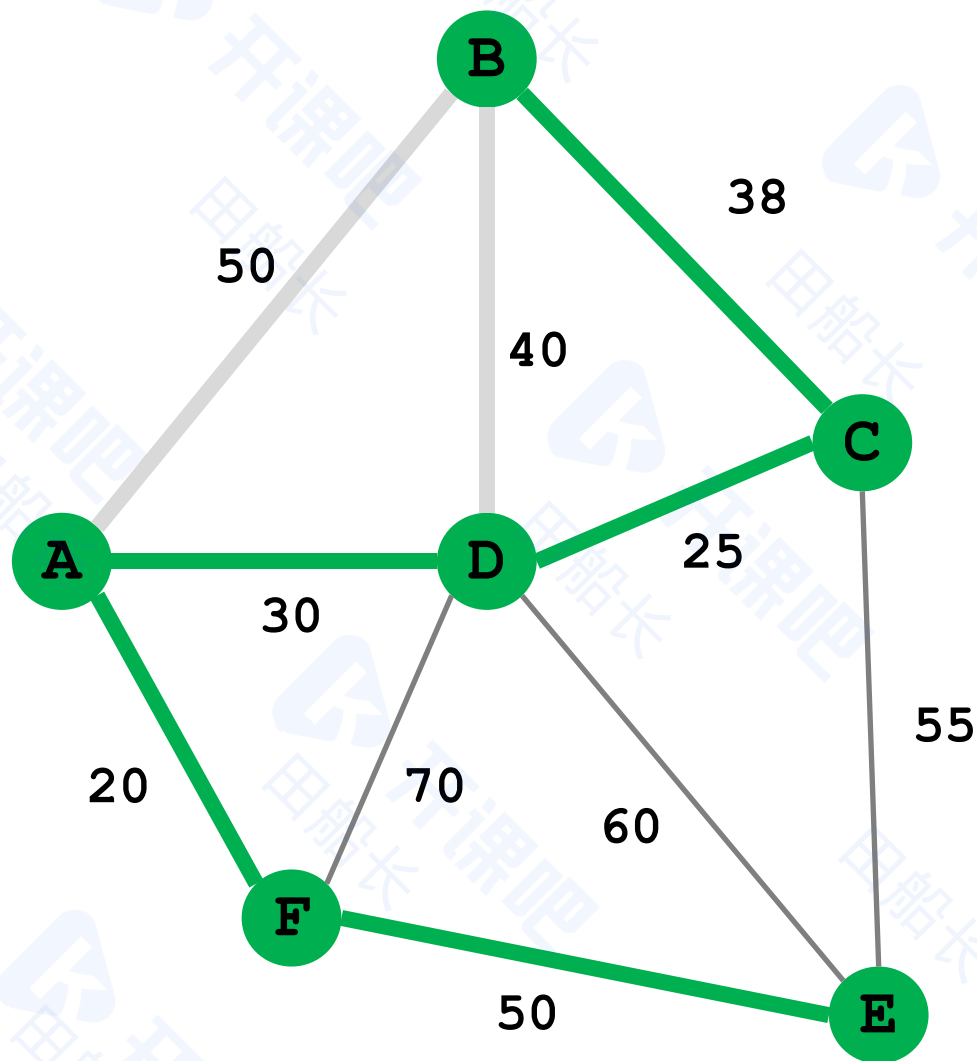


# Kruskal 算法



这条边的两个端点不相连  
选中这条边  
连通这两点

# Kruskal 算法



共有 $n$ 个点  
选出了 $n-1$ 条边  
此时所有点均已连通  
最小生成树求解完毕