图应用

优先级搜索

各种遍历算法在本质上都是类似的，其区别仅仅在于选取访问顶点的次序。

广度：优先访问与更早被发现的顶点，考察与其邻接的节点。

深度：优先访问与更晚被发现的顶点，考察与其邻接的节点。

优先级搜索：给定某个优先级进行搜索。

$O(n^2)$，采用优先级队列可以优化到 $O((e+n)\log n)$.

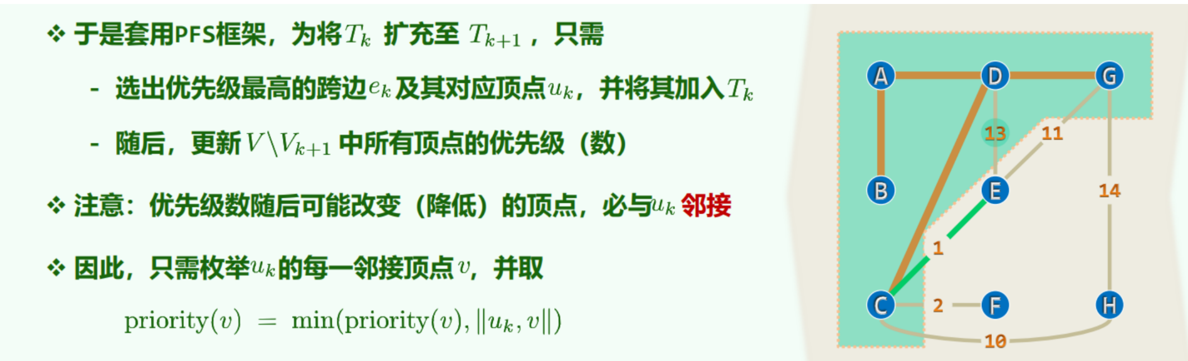
最小生成树

如果有负权值，可以统一进行调整，因为边数是确定的。

可能会有多个最小生成树，没关系。

Prim 算法

每次找到当前的最短跨边，加入其中并更新其他点到当前树的距离



思路与dijkstra很类似，只是prim顶点标号是树边，优先级是到当前树的距离；dijkstra点优先级是到顶点的距离。

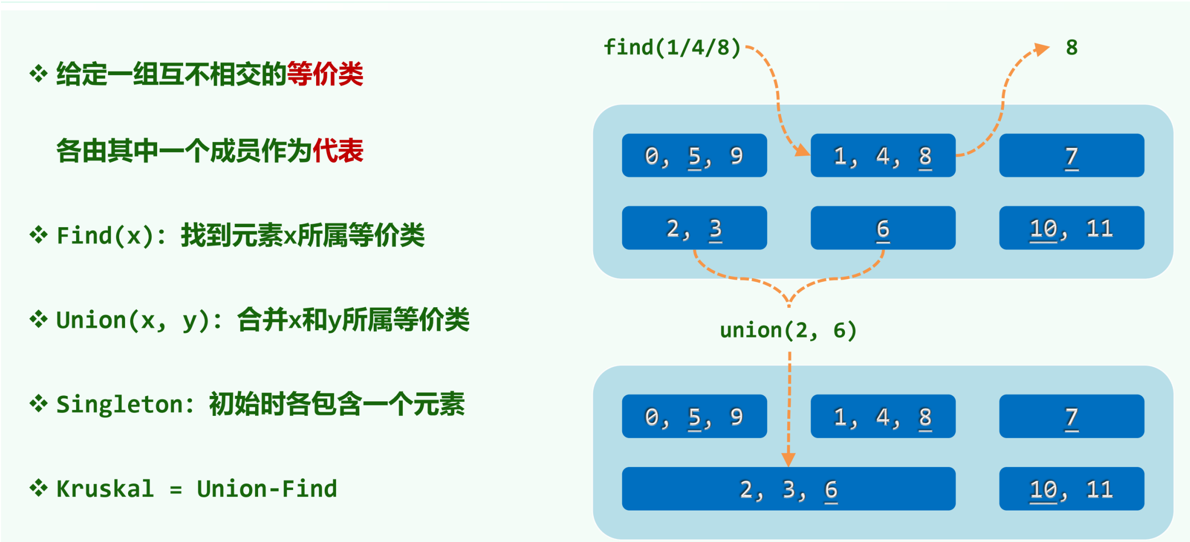
Kruskal 算法

贪心规则

每次加入最短边，若成环则舍弃

并查集

动机：方便Kruskal森林的查找与合并

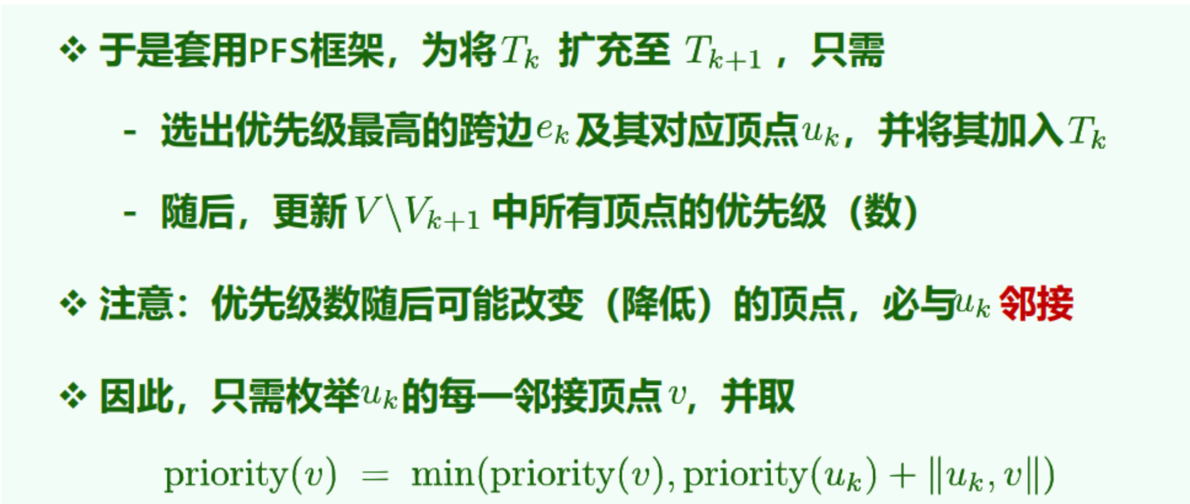


最短路

Dijkstra 算法

E. Dijkstra, 1959. Single Source Shortest Path.

维护已访问的点集，每次将与点集相邻且到顶点距离最短的点加入这个点集，并依据这个点更新其他点的最短距离 。



Floyd Warshall 算法

Floyd Warshall, 1962.

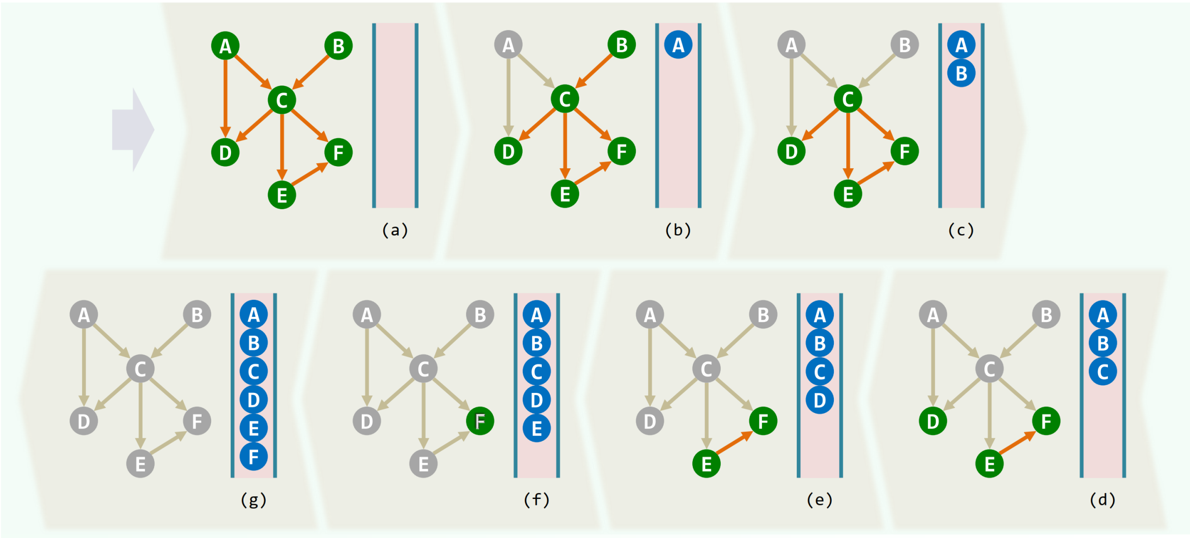
以上内容见离散数学整理。

拓扑排序

任给一个图G，尝试将其所有点线性排列，与原图相容，并保证不会有从某一点触发指向其前驱顶点的边。

零入度算法

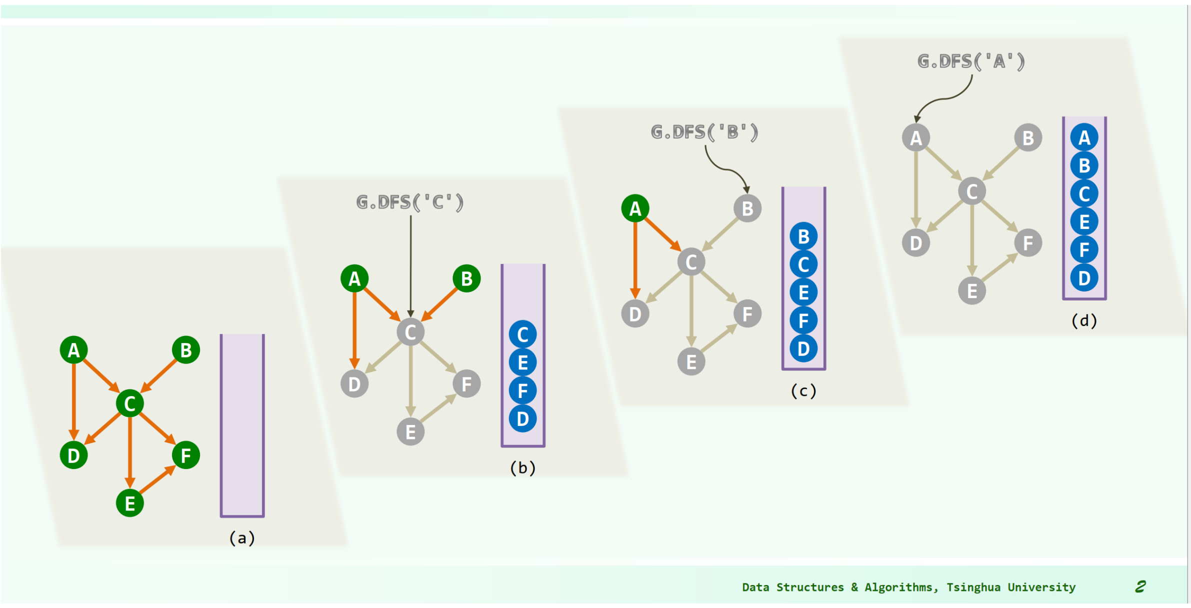
首先线性扫描所有顶点，入度为0的结点入栈。随后每次取出这些栈顶元素入队，删除从这个点出发的边，并将新的零入度结点入栈。如此重复直至栈空且图空，最后队列中得到的就是一个拓扑排序。



不存在拓扑排序的条件是算法终止时图非空

零出度算法

对图DFS，访问节点依次入栈，最后得到的所有点满足按照ftime降序排列，也就是拓扑排序



不存在拓扑排序的条件是DFS时发现后向边

双联通分量

概念

对于一个无向图 G。如果删掉节点 v 之后之后连通块增多，称之为割点/关节点。

无割点的图，称之为双连通图。

极大的双联通子图，称之为双联通分量。

算法

用 dfs 判定双联通分量。考察 dfs 生成的 dfs 树。无向图的 dfs 树有一个很好的性质，它不会有cross边，只会有backward边，所以可以考察 dfs 树的 `subtree(v)`，它往上走的边只会到 v 到根的路径上。

算法的关键在于找出关节点。

- DFS树上的叶子节点一定不可能是关节点。
- DFS树上的非叶节点：
    - 不是关节点的充分必要条件：
        - 至少有两棵子树
        - 所有子树能够到达的最高的祖先都在 v 以上。
    - 如果是关节点，那么关节点和子树可以构成一个双联通分量。

时间复杂度：$O(n+e)$