并查集

主讲：肖臻

并查集（Union-Find）

- 教材的相关章节
  - Algorithm Design，Chap4.6，英文版p151
  - Intro to Algo，2nd edition，Chap21，英文版p498
- 工作原理
  - 数据结构是棵倒着长的树
  - 元素两两做Union操作，k次union最多涉及到2k个元素
  - 每次Union后，不同集合的数目恰好减一
    - 假设做Union的两个元素原属于不同集合
    - 全部合并为一个大集合恰好需要n-1次Union操作

性能优化

- Path compression
- Link by size: Algorithm Design
  - 只有根节点才会更新size或rank信息
- Union by rank: 《算法导论》
  - 《算法导论》中定义的rank是树的高度的上界
    - 这个上界没有在路径压缩过程中调整，主要是用于理论分析
- 实际应用中，只实现路径压缩往往就够了
  - 如果应用不需要size或rank信息的话，不一定要维护：不影响正确性（比如蝴蝶分类问题）
  - 注意：只有根节点的size才有意义，因为路径压缩会使其它节点的size信息过时

性能分析与应用

- 性能分析
  - log*n：取多少次log之后，值变成1？
- 并查集的应用
  - MST：Kruskal算法
    - 主要性能瓶颈是初始时的排序
  - 图的连通分量
  - 第一次作业：The Suspect、蝴蝶分类
- 有些应用需要维护除了rank之外的其它信息
  - 比如查找元素所在集合的最大值
  - 有些应用需要creativity

并查集的适用范围

- 并查集的局限
  - 不支持split操作：任何节点一旦成为其它节点的子节点后，将永远不可能再成为树根
  - 集合必须是disjoint：同一个元素不能属于多个集合
  - 集合代表不记录集合成员信息
    - 父节点不记录子节点的信息
    - 比如查找图中某节点所在的连通分量的所有节点需要再次扫描或者维护额外信息
- Successor with delete
  - 给定N个整数，0,1,…,N-1，需要支持下面两个操作：（1）删除元素x（2）找到x的后序元素

并查集扩展

- 每对元素之间不一定是等价关系
  - 蝴蝶分类的例子：Chap3-Ex4，英文版p107
  - 难点：等价关系具有传递性，但是不等价关系则没有传递性并且与集合总数目有关
  - 解决方法：记录到父节点的“偏移量”
    - 蝴蝶分类：偏移量为binary，构成回路的三角形或四边形偏移量之和必须为偶数
- 思考题：并查集的应用都可以用遍历图（BFS、DFS）的方法来解，那还要并查集何用？
  - Dynamic connectivity：判断图的最早连通时间
  - Percolation：no closed form solution