

并查集



分离集合

- 在有的问题中，需要对不相交的集合(disjoint set)进行这样两种操作：
 - 检索某元素属于哪个集合
 - 合并两个集合
- 能够维护这两个操作的数据结构，我们称之为并查集。

并查集的森林实现

- 一般来说我们用森林的结构实现并查集
- 在森林中，每棵树代表一个集合。用树根来表示这个集合。
- 合并操作：两个集合**S1**、**S2**合并，将其中的一个树根作为另一个树根的子树即可。
- 查找操作：对于一个元素u的查找，顺着u往上找，直到线索到根节点，也就确定了u所在的集合。

两个优化

- 启发式合并：

在合并集合S1、S2的时候，我们让较小的树成为较大的树的子树。这里可以是深度、节点个数等启发函数来比较树的大小。

- 路径压缩：

我们在查找完u至根节点的路径之后，一般将这条路径上的所有节点的父节点都设为根节点，这样可以大大减少之后的查找次数。

并查集的时间复杂度

- 可以证明，经过启发式合并和路径压缩之后的并查集，执行 m 次查找的复杂度为 $O(m\alpha(m))$
- 其中 $\alpha(m)$ 是Ackermann函数的某个反函数，你可以近似的认为它是小于5的。所以并查集的单次查找操作的时间复杂度也几乎是常数级的。