# 总结

■ 摘自《维基百科》： https://en.wikipedia.org/wiki/Disjoint-set_data_structure#Time_complexity

> Using both *path* ***compression***, ***splitting***, *or* ***halving*** and *union by* ***rank*** *or* ***size*** ensures that the amortized time per operation is only $O(\alpha(n))$,[4][5] which is optimal,[6] where $\alpha(n)$ is the inverse Ackermann function. This function has a value $\alpha(n) < 5$ for any value of $n$ that can be written in this physical universe, so the disjoint-set operations take place in essentially constant time.

■ 大概意思是

□ 使用路径压缩、分裂或减半 + 基于rank或者size的优化

✓ 可以确保每个操作的均摊时间复杂度为 $O(\alpha(n))$，$\alpha(n) < 5$

■ 个人建议的搭配

✓ Quick Union

✓ 基于 rank 的优化

✓ Path Halving 或 Path Spliting

小码哥教育 SEEMYGO

# 自定义类型

- 之前的使用都是基于整型数据，如果其他自定义类型也想使用并查集呢？
  - 方案一：通过一些方法将自定义类型转为整型后使用并查集（比如生成哈希值）
  - 方案二：使用链表+映射（Map）