

Brief Probe Into Binary Lifting

LCA 与 RMQ 入门

主讲: 赵涵铮

agile studio

February 10, 2023

目录

1 引言

- LCA、RMQ 简介
- 树的祖先

2 LCA 算法

- 朴素算法
- 倍增算法

3 RMQ 算法

目录

1 引言

- LCA、RMQ 简介
- 树的祖先

2 LCA 算法

- 朴素算法
- 倍增算法

3 RMQ 算法

LCA、RMQ 简介

- LCA 是 Least Common Ancestors 的缩写，即为最近公共祖先，用于寻找两个点的最近公共祖先
- RMQ 是 Range Maximum Query 的缩写，即区间最值查询，用于回答 (i, j) 之间的最值。
- 以上算法都有基于倍增的做法。

树的祖先

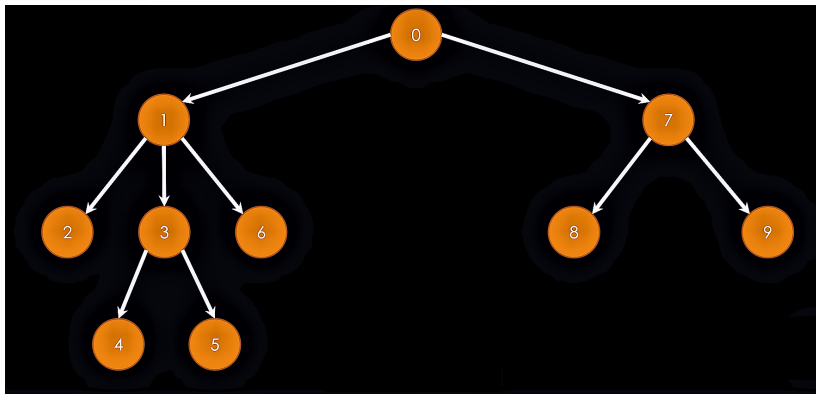


Figure: 树

目录

1 引言

- LCA、RMQ 简介
- 树的祖先

2 LCA 算法

- 朴素算法
- 倍增算法

3 RMQ 算法

暴力小跳——朴素算法

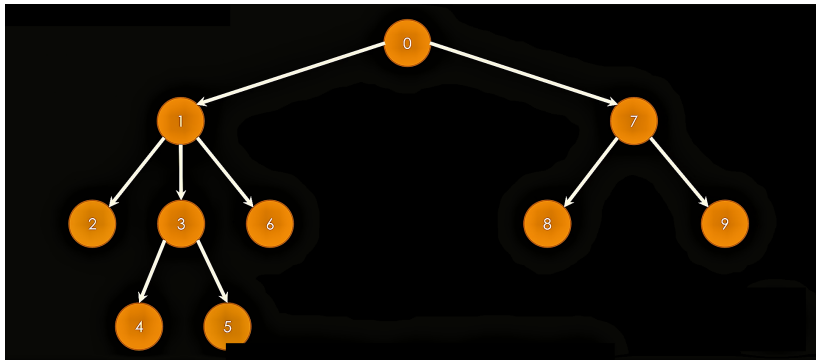
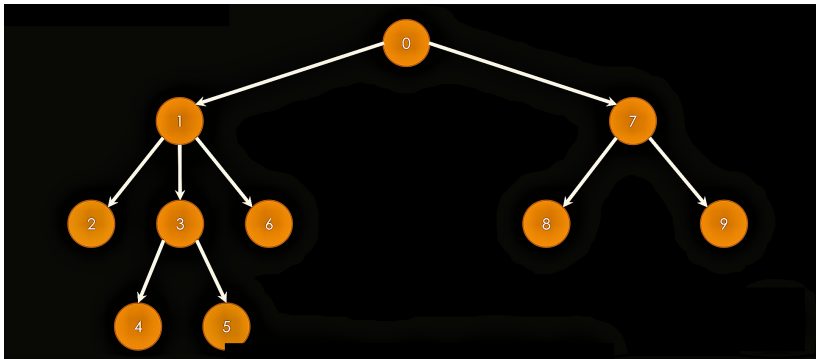


Figure: 如何寻求 2 和 5 的公共祖先呢？

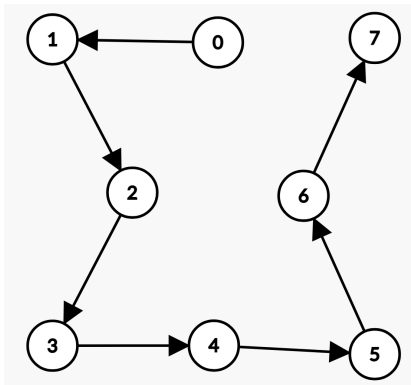
暴力小跳——朴素算法

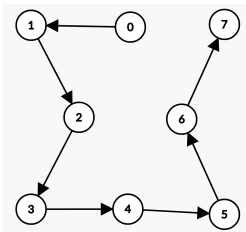
- 将 5 跳到与 2 相同的深度
- 两个点同时向上跳
- 当遇到链状结构时，时间会很慢，退化到 $O(N)$



大跳——倍增算法

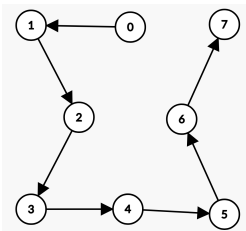
为了跑的更快、跳得更远，我们需要维护每个节点的多级祖先，而非直接祖先。以一条链为例





如果我们将节点 7 的所有祖先都维护上，那么查询的时间复杂度是 $O(1)$ 的，但是预处理和空间复杂度是 $O(N^2)$ 的，这是我们无法接受的，为了在时间和空间上进行权衡，保留节点 2^i 次方级祖先的信息。

维护多级祖先



1 级祖先	6
2 级祖先	5
4 级祖先	3

Table: 节点 7 的多级祖先

我们用 $fa[i][j]$ 去表示节点 i 的 2^j 级祖先，上表即表示为：

$$fa[7][0] = 6, fa[7][1] = 5, fa[7][2] = 3$$

目录

1 引言

- LCA、RMQ 简介
- 树的祖先

2 LCA 算法

- 朴素算法
- 倍增算法

3 RMQ 算法

倍增在 RMQ 上的应用——st 表

题目描述

给定一个长度为 N 的数列，和 M 次询问，求出每一次询问的区间内数字的最大值。

假设我们要求下列数字中下标 $2 \rightarrow 6$ 的最大值

4 8 7 2 1 3 6 9 5

由于询问长度为 5，我们只要预处理求出从下标 2 开始询问长度为 4 的最大值、从下标 3 开始长度为 4 的区间最大值即可。
与倍增求 lca 如出一辙。

Thank you

Thank you for listening!

