1 P24 Splay Trees 势能分析

现在我们讨论使用势能方法分析 splay 伸展树的均摊成本。

分析的关键是要找到合适的势能函数。这也是很多势能分析问题的难点所在。我们注意到 Splay 操作中，会将所访问的节点翻转到根节点。前面的样例中我们也注意到，翻转代价高的操作往往更大程度上的降低了树高（比如前面例子中，将节点 1 从叶节点位置翻转到根节点，大致将树高缩减为原来的一半）。所以我们考虑一个跟节点高度相关的（或类似的）势能函数。

我们注意到在 Splay 操作中，几乎每个节点的高度都会改变，哪怕该节点为根节点的子树没有任何变化。如果我们直接使用节点高度作为势能函数，后续的数学计算与推导会变得非常复杂。我们希望有一个可以简化后续数学推导的势能函数。

一个可用的势能函数是树中所有节点的 rank 之和：

$$\Phi(T) = \sum_{i=1}^{n} \log S(i)$$

其中 $S(i)$ 指的是子树 i 中的节点数（包括节点 i）。我们用 $R(i)$ 表示节点 i 的 rank，$R(i) = \log S(i)$。

2 P25 Splay 势能分析

势能函数 $\Phi(T) = \sum_{i \in T} R(i)$。对于我们刚才学过的 3 种 splay 操作：zig、zig-zig、zig-zag，我们使用 R_2 表示操作后的势能，R_1 表示操作前势能。首先是 zig 操作，做了个单旋，成本为 1；从 PPT 上的 zig 操作示意图中可以看出，在整个操作中只有 X 和 P 的 rank 值有变化。所以我们有：

$$\hat{c_i} = 1 + R_2(X) - R_1(X) + R_2(P) - R_1(P)$$

由于节点 P 由根节点变为非根节点，我们有 $R_2(P) - R_1(P) \leq 0$，因此 $\hat{c_i} \leq 1 + R_2(X) - R_1(X)$。既然 $R_2(X) - R_1(X) \geq 0$，我们有 $\hat{c_{i-zig}} \leq 1 + 3(R_2(X) - R_1(X))$。

对于 zig-zag 操作，实际成本是两次旋转，为 2。因此，

$$\hat{c_i} = 2 + R_2(X) - R_1(X) + R_2(P) - R_1(P) + R_2(G) - R_1(G)$$

从左图中看到，操作前 G 是根节点，操作后 X 是根节点，他们的 rank 相同，因此，$\hat{c_i} = 2 - R_1(X) + R_2(P) - R_1(P) + R_2(G)$。同时，操作后我们可以看到 $S_2(P) + S_2(G) \leq S_2(X)$，由书上的 lemma 11.4，我们可以得出：$R_2(P) + R_2(G) \leq R_2(x) - 2$。故我们有：

$$\hat{c_i} \leq 2(R_2(X) - R_1(X))$$

zig-zig 的推导就留给同学们自己完成。

最后，给定一个伸展树上访问节点 X 的一系列 M 个 splay 操作（zig、zigzig、zigzag），其中最多只会有 1 个 zig。把他们都给加起来后，可得：

$$\sum_{i=1}^{M} \hat{c_i} \leq 3(R_M(X) - R_1(X)) + 1$$

注意到结束操作后，X 是新的根节点，所以 $R_M(X) = R_1(T)$，所以我们有

$$\sum_{i=1}^{n} \hat{c_i} \leq 3(R_1(T) - R_1(X)) + 1$$

很明显，均摊成本是 $O(\log n)$ 级别的。