大家请对照课程 PPT 的第 11 页和 12 页阅读下面材料。

1 势能分析方法回顾

首先回顾下均摊分析的势能方法：

D_i：执行第 i 步操作后的数据结构；

D_0：初始数据结构；

势能函数：$\Phi : D_i \to R$，反应操作后数据结构的势能；

c_i: 将 $D_i - 1$ 变换到 D_i 的实际成本；

$\hat{c}_i$: 将 $D_i - 1$ 变换到 D_i 的均摊成本，我们有：$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$；所以，总均摊成本为：$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} c_i + \Phi(D_n) - \Phi(D_0)$

在这个总均摊成本的计算中，如果我们能够保证 $\Phi(D_n) - \Phi(D_0) \geq 0$，我们就能保证总均摊成本是总实际成本的一个上限。势能方法的关键是找到合适的势能函数。一般一个好的势能函数设计总能使 $\Phi(D_0)$ 是这个序列中的最小值。实际使用中，我们一般定义 $\Phi(D_0) = 0$。这样，只要证明上式中 $\Phi(D_i) \geq 0$，就可以证明均摊成本是实际成本的一个上限。

2 斜堆合并的均摊分析

斜堆合并的总步骤数就是两个堆的右路径节点数之和；但是斜堆右路径长度不受限，最坏情况可到 $O(n)$，所以我们需要通过均摊分析来确定 M 个操作序列的复杂度。

均摊分析的势能函数要反映斜堆合并操作的效果。考虑到合并成本为右路径总节点数，直接想法是通过右路径节点数目来定义势能函数（斜堆合并由空堆开始，因此该函数刚好从 0 开始且非负）。但该函数的问题是单调递增的，不能反映合并过程中的势能变化。

我们选择斜堆中的“重节点（heavy nodes）”数作为势能函数。如果一个节点的右子树总结点数占该节点为根的子树总结点数达一半及以上为重节点，反之为轻节点。注意到斜堆合并是由空堆开始，该势能函数满足：$\Phi(D_0) = 0$；随着合并的进展，中间任何步骤都有 $\Phi(D_0) \geq 0$。

假设要合并的两斜堆 H_1 和 H_2 右路径上的重节点数分别为 h_1 和 h_2。俩斜堆中其他重节点数目为 $\mathbf{h}$。则有：

$$\Phi(D_0) = h_1 + h_2 + \mathbf{h}$$

注意在合并过程中，除了右路径节点，其他节点不会发生轻重转变，因为它们的左右子树都没有变化。

因为合并成本为右路径总节点数，如果 H_1 和 H_2 右路径上的轻节点数分别为 l_1 和 l_2，则实际合并的总代价为：

$$\sum_{i=1}^{n} c_i = l_1 + l_2 + h_1 + h_2$$

合并后，只有右路径节点会出现轻重节点的改变：（1）右路径重节点合并后会肯定变成轻节点（左右调换后，左轻右重自然变成了左重右轻）；（2）右路径轻节点合并后有可能变成重节点。所以合并后重节点数目最多的情况就是所有右路径的轻节点都变重节点的情况。所以：$\Phi(D_N) \leq l_1 + l_2 + \mathbf{h}$。所以：$(\Phi(D_N) - \Phi(D_0)) \leq l_1 + l_2 - h_1 - h_2$。所以：

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} c_i + (\Phi(D_N) - \Phi(D_0)) \leq 2(l_1 + l_2)$$

l_1 和 l_2 为要合并的俩斜堆右路径上轻节点数量。轻节点意味着节点的子树“左重右轻”（和前面讲过的左倾堆类似），因此 $l_1 + l_2 \leq \log N_1 + \log N_2$（$N_1$ 和 N_2 分别为俩斜堆的节点数）。