数据结构与算法 I 思考题 9

2019201409 于倬浩

2021 年 1 月 2 日

问题

假设我们将斐波拉契堆的 decrease-key 操作改为：一个节点在失去 k 个孩子后才会被切开，其中 k 为大于 2 的常数。证明这样的改动使得 decrease-key 的摊还代价降低常数倍，而 extract-min 的摊还代价增加常数倍。

证明

修改势函数的定义。设 $F_i(D_j)$ 表示斐波那契堆 D_j 中，被标记 i 次的节点个数。那么，势函数的计算方式修改如下：

$$\phi(D_i) = \sum_{j=1}^{k}(k-j+1)F_j(D_i) + t(D_i)$$

那么，每次标记某个节点，会导致势函数增加 1，或是减少 k。一次 decrease-key 操作，假设调用了 c 次 cut，则必定减少了 $c-1$ 个标记次数为 k 的节点，因此势函数必定减少 $-k\times(c-1)$，另外，最多标记了 c 次节点，因此又带来 c 的增量。另外，由于又增加了 $c-1$ 个根，因此势函数改变量最大为：

$$-k\times(c-1)+c-1 = k-1-(k-1)c$$

均摊代价最多为 $O(c)+k-1-(k-1)c = O(c)+k-1 = O(c)$，对比原算法的均摊代价为 $O(c)+4-c = O(c)$。由于第一个 $O(c)$ 中的常数因子没有改变，而当 $k>2$ 时，$k-1<1$，因此实际上减去的部分中常数因子更大，所以 decrease-key 操作会有常数上的改进。另外，尽管实际上增加了一个 $k-1$ 的常数因子，由于减去的部分 $(k-1)c$ 还有一个系数 c，所以增加的常数因子比减少的要小，即实际上会有常数上的改进。

另一方面，由于 extract-min 操作中，最多有一个节点的标记数量增加，由于新的势函数保证了增量和原有算法势函数保持一致，因此实际上在这部分的势能分析和原算法类似，不再赘述，均摊代价为 $O(D(n))$，仅需考虑 $O(D(n))$ 的变化即可（$D(n)$ 为节点度数）。

重新考虑我们对于节点度数的下界分析：假设当前有一个度数为 k 的节点 x，现在我们正在插入第 i 个节点 y_i，此时保证 $x.degree \geq i-1$, $y_i.degree \geq i-1-(k-1) = i-k$，和原算法的

$y_i.degree \geq i-2$ 不同。对于其余部分，分析思路和 CLRS 上 *Lemma* 19.1 到 *Lemma* 19.3 保持一致，唯一不同的是，由于 $y_i.degree >= 0$ 的条件变为了 $i >= k$，因此实际上递归式不再是严格的斐波那契数列，而是需要修改第 0 到第 $k-1$ 项为常数，之后从第 k 项开始才能继续沿用原有递推式。使用《离散数学》的知识分析，得知这样虽然可以保证节点度数 $k \leq \lfloor log_{\phi'} n \rfloor$，但是实际上这里的 ϕ' 需要比原来的 ϕ 小，底数变小，基数不变，因此带来常数上的增加，即节点度数的上界增加一个常数（特征方程不变，通解不变，但由于初始条件改变，因此得到的解发生变化）。具体分析过程与本门课程无关，在此略去。

因此结论是，对于 `decrease-key` 操作，修改后的算法会减小常数代价，而对于 `extract-min` 操作，修改后的算法会导致摊还代价增加常数倍。