算法基础第二次作业

习题 1

假设使用的是大顶堆，且此堆有 n 个节点，则高度为 $h = \lfloor \log_2 n \rfloor$。

升序 A 和降序 A 的根本区别是：降序的 A 本身就是一个堆，而升序 A 所有的非叶子节点为根形成的子树都不满足堆的定义，且只有建堆时的时间复杂度存在区别。在堆排序的建堆后阶段无论是什么顺序都要经历 $n-1$ 次时间复杂度为 $\Theta(\lg n)$ 的 MAX-HEAPIFY 环节，总的时间复杂度是 $\Theta(n \lg n)$。

升序 A 建堆过程的代价是

$$S_n = \sum_{h=0}^{\lfloor \log_2 n \rfloor} \lceil \frac{n}{2^{h+1}} \rceil O(n) = O(n \sum_{h=0}^{\lfloor \log_2 n \rfloor} \frac{h}{2^h}) = O(n \sum_{h=0}^{\infty} \frac{h}{2^h}) = O(n)$$

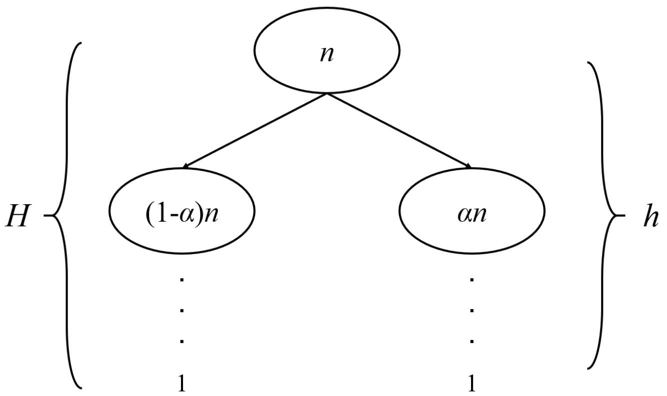
降序 A 建堆过程的代价是

$$S_n = \frac{n}{2} = O(n)$$

两种情况的建堆代价都不超过排序的代价,所以二者的时间复杂度均是 $\Theta(n \lg n)$。

习题 2

（a）对 PATITION 问题画出递归树，每一个节点上的数表示这次需要对多少个数进行操作。设最大深度为 H，最小深度为 h。之所以左侧是 H 是因为 $0 < \alpha \leq 1/2$。



因为不考虑舍入，粗略计算：

$$(1-\alpha)^H n = 1 \quad \alpha^h n = 1$$

于是

$$H = -\lg n / \lg(1-\alpha) \quad h = -\lg n / \lg \alpha$$

（b）假设要产生一个比 $1-\alpha$ 更不平衡的划分，PATITION 选择的主元需要落在最小的 αn 个数中或最大的 αn 个数中，此概率约为

$$\bar{p}=\frac{\alpha n+\alpha n}{n}=2\alpha$$

所以所求概率约为

$$p=1-\bar{p}=1-2\alpha$$