Design and Analysis of Algorithms

Tutorial 4: Sorting and Searching

童咏昕

北京航空航天大学 计算机学院

yxtong@buaa. edu. cn

问题1

- 课上讨论了时间复杂度为$O(n\ log\ n)$的建立最小堆(min-heap)的算法，请设计一个时间复杂度为$O(n)$的建立最小堆的算法。

问题1-提示

- 对数组建立完全二叉树(结构与最终堆的结构完全相同，只是不符合最小堆的性质)。
  该二叉树可完全使用初始数组进行操作，不耗费时间。
- 对二叉树从低层到高层(叶节点为第一层而根节点为最高层)，对每一层的节点进行bubble down操作，以使其满足最小堆的性质。
  假设二叉树高度为h，第i层中，至多有2^{h-i}个节点，每个节点进行bubble down操作，至多需要O(i)的时间复杂度(递归至叶节点)，因此该层的时间复杂度为$O(i \cdot 2^{h-i})$.

问题1-提示

二叉树高度$h = \log_2 n$, 因此总的运行时间为

$$
\begin{aligned}
&\mathrm{O}(n) + \sum_{i=1}^{h} \left(O\left(i \cdot 2^{h-i}\right) + O(1)\right) \\
&= \sum_{i=1}^{\log_2 n} c_1 \cdot i \cdot 2^{\log_2 n - i} + c_2 \log_2 n \\
&= c_1 n \sum_{i=1}^{\log_2 n} \frac{i}{2^i} + c_2 \log_2 n \\
&\le c_1 n \sum_{i=1}^{\infty} \frac{i}{2^i} + c_2 \log_2 n \\
&= 2c_1 n + c_2 \log_2 n \\
&= O(n)
\end{aligned}
$$

问题1-提示

其中倒数第二步由下式得到

$$\sum_{i=1}^{\infty} \frac{i}{2^i} = \sum_{i=1}^{\infty} \sum_{j=i}^{\infty} \frac{1}{2^j} = \sum_{i=1}^{\infty} \frac{1}{2^{i-1}} = 2$$

问题2

- 给出算法支持最小堆上的Decrease-Key操作，要求时间复杂度为$O(\log n)$。

  Decrease-Key:减小堆中某个元素的值。

问题2-提示

- 注意到最小堆中，某个节点A的父亲结点总是小于A的儿子节点，因此在将某元素的值减小之后，只需要在该节点到根节点的路径中，依次将不符合最小堆性质的一对节点的值(一定为父子关系)交换即可。

Heap-Decease-Key($heap, node, newKey$)

```
Input: heap is the original heap, node is the node waiting to be
       decreased, newKey is the new value of node
Output: The new heap heap
if newKey > node.key then
  | return error;
end
node.key ← newKey;
while node ≠ Root(heap) and (Parent(node).key > node.key) do
  | swap node.key and Parent(node).key;
  | node ← Parent(node);
end
return heap;
```

问题3

- 帽子核对问题：n位顾客，他们每人给餐厅服务生一顶帽子。服务生以随机顺序将帽子归还给乘客，求拿到自己帽子的顾客数量的期望。

问题3-提示

- 定义如下指示器变量

$$X_i = \begin{cases} 1, & \text{当第i名顾客拿到自己的帽子} \\ 0, & \text{否则} \end{cases}$$

则有$E(X_i) = \Pr(X_i = 1) = \frac{1}{n}$

$$E(X) = E\left(\sum_{i=1}^{n} X_i\right)$$

$$= \sum_{i=1}^{n} E(X_i) = \sum_{i=1}^{n} \frac{1}{n} = 1$$

注意：各个X_i之间不相互独立，例如，当$X_1 = X_2 = \cdots = X_{n-1} = 1$时，$X_n$一定为1.

但是这不影响上述推导。