

算法导论习题选集

作业 9

节选自《算法导论》教材第三版

课程网站：<https://algorithm.cuijiacai.com>

Problem 1

(利用链表实现可合并堆) 可合并堆 (**Mergeable Heap**) 支持以下操作: MAKE-HEAP (创建一个空的可合并堆)、INSERT、MINIMUM、EXTRACT-MIN 和 UNION。说明在下列前提下如何利用链表实现可合并堆。试着使各种操作尽可能高效。分析每个操作关于动态集合规模的运行时间。

1. 链表是已排序的。
2. 链表是未排序的。
3. 链表是未排序的, 且待合并的动态集合是不相交的。

Problem 2

(搜索已排序的紧凑链表) 练习 9-2 问题 2 讨论了如何将 n 个元素的链表紧凑地维持在数组的前 n 个位置。假设所有的关键字均不相同, 且紧凑链表是已排序的, 即对所有的 $i = 1, 2, \dots, n$ 且 $next[i] \neq NIL$, 有 $key[i] < key[next[i]]$ 。又假设有一个变量 L 存放链表的首元素的下标。在这些假设下, 试说明可以利用下列随机算法在 $O(\sqrt{n})$ 的期望时间内搜索链表。

```
COMPACT-LIST-SEARCH( $L, n, k$ )
1   $i = L$ 
2  while  $i \neq NIL$  and  $key[i] < k$ 
3       $j = \text{RANDOM}(1, n)$ 
4      if  $key[i] < key[j]$  and  $key[j] \leq k$ 
5           $i = j$ 
6          if  $key[i] == k$ 
7              return  $i$ 
8       $i = next[i]$ 
9  if  $i == NIL$  or  $key[i] > k$ 
10     return  $NIL$ 
11 else return  $i$ 
```

如果忽略过程中第 3 到 7 行, 就得到一个普通的搜索已排序链表的算法, 其中下标 i 依次指向链表的各个位置。当下标 i 越出表的末端或 $key[i] \geq k$ 时, 搜索终止。在后一种情况中, 如果 $key[i] = k$, 显然, 我们已找到值为 k 的关键字。但如果 $key[i] > k$, 则我们永远也找不到值为 k 的关键字, 因而终止查找是正确的。

第 3 到 7 行意图向前跳至某个随机选择的位置 j 。当 $key[j]$ 大于 $key[i]$ 而不大于 k 时, 这种跳跃是有益的。因为这种情况下, j 在链表中标识了一个正常搜索中 i 将要到达的位置。由于该链表是紧凑的, 所以在 1 到 n 中任意选择一个 j 都会指向链表中的某个对象, 而不会是自由表中的某个位置。

我们不直接分析 COMPACT-LIST-SEARCH 的性能, 而是要分析一个相关的算法 COMPACT-

LIST-SEARCH'，该算法执行两个独立的循环。该算法增加了一个参数 t ，用来决定第一个循环迭代次数的上限。

```

COMPACT-LIST-SEARCH'(L, n, k, t)
1  i = L
2  for q = 1 to t
3      j = RANDOM(1, n)
4      if key[i] < key[j] and key[j] ≤ k
5          i = j
6          if key[i] == k
7              return i
8  while i ≠ NIL and key[i] < k
9      i = next[i]
10 if i == NIL or key[i] > k
11     return NIL
12 else return i

```

为了比较算法 COMPACT-LIST-SEARCH(L, n, k) 和 COMPACT-LIST-SEARCH'(L, n, k, t) 的执行过程，假定调用 RANDON($1, n$) 所返回的整数序列在两个算法中是一样的。

1. 假设 COMPACT-LIST-SEARCH(L, n, k) 中第 2 到 8 行的 **while** 循环经过了 t 次迭代。论证 COMPACT-LIST-SEARCH'(L, n, k, t) 会返回同样的结果, 且 COMPACT-LIST-SEARCH' 中的 **for** 循环和 **while** 循环的迭代次数之和至少为 t 。

在 COMPACT-LIST-SEARCH'(L, n, k, t) 的调用中，设随机变量 X_t 描述了第 2 到 7 行的 **for** 循环经 t 次迭代后链表中从位置 i 到目标关键字 k 所在位置之间的距离（即通过 *next* 指针链的次数）。

2. 论证 COMPACT-LIST-SEARCH'(L, n, k, t) 的期望运行时间为 $O(t + E[X_t])$ 。

3. 证明: $E[X_t] \leq \sum_{r=1}^n (1 - r/n)^t$ 。提示: 当随机变量 X 可在自然数集 $\mathbb{N} = \{0, 1, 2, \dots\}$ 中取值时，有一个很好的期望计算公式:

$$\begin{aligned} E[X] &= \sum_{i=0}^{\infty} i \cdot \Pr\{X = i\} = \sum_{i=0}^{\infty} i \cdot (\Pr\{X \geq i\} - \Pr\{X \geq i+1\}) \\ &= \sum_{i=1}^{\infty} \Pr\{X \geq i\} \end{aligned}$$

4. 证明: $\sum_{r=0}^{n-1} r^t \leq n^{t+1}/(t+1)$ 。
5. 证明: $E[X_i] \leq n/(t+1)$ 。
6. 证明: COMPACT-LIST-SEARCH'(L, n, k, t) 的期望运行时间为 $O(t + n/t)$ 。
7. 证明: COMPACT-LIST-SEARCH 的期望运行时间为 $O(\sqrt{n})$ 。
8. 为什么要假设 COMPACT-LIST-SEARCH 中的所有关键字均不相同? 论证当链表中包含重复的关键字时, 随机跳跃不一定能降低渐近时间。

(续页)

(续页)