

Algorithm Homework1

PB18111704 Zhu Enzuo

2020 年 10 月 2 日

0.1 Prob1

a1 线性查找伪代码如下

Algorithm 1 线性查找

```
1: procedure FETCH( $A, v$ )  
2:   for  $i \leftarrow 1, n$  do  
3:     if  $a_i = v$  then  
4:       return  $i$   
5:     end if  
6:   end for  
7:   return  $NIL$   
8: end procedure
```

a2 循环不变式证明如下

初始化 首先需要证明迭代开始之前循环不变式成立。此时 $i=0$ ，已查找的元素集合中和 v 不相等的元素的集合 S 为 \emptyset 。而空集中不会有元素和 v 相等。

保持 我们需要证明每一次循环之后 $v \notin S$ ，而我们可以看到只有满足 $a_i \neq v$ 的 a_i 才会被加入到 S 当中。故该性质对循环成立。

终止 当循环中途跳出的时候,说明我们找到了一个 $a_i = v$, 满足要求。当循环正常结束之后,说明对任意 a_i , 都有 $a_i \neq v$, 满足题目要求。故我们可以得出结论该算法是正确的。

b 平均需要检查的元素个数 $P = \sum_{k=1}^n \frac{k}{n} = \frac{k+1}{2}$, 最坏情况需要检查的元素个数为 n 。 $f(A, v) = O(n) = \Theta(n)$

0.2 Prob2

a 错误。考虑 $f(n) = \frac{1}{n}$, 可以发现 $O(f(n)^2) = \frac{1}{n^2} < f(n)$ 。

b 正确。利用 Θ 的定义, 取 $c_1 = 1, c_2 = 2, n_0 = 1$, 有 $\forall n \geq n_0, 0 \leq \max(f(n), g(n)) \leq f(n) + g(n) \leq 2\max(f(n), g(n))$ 。故命题正确。

c 错误。取 $f(n) = n, O(f(n)) = n^2$, 有 $\Theta(f(n)) = n$, 显然 $f(n) + O(f(n)) = n^2 \neq \Theta(f(n))$ 。

d 错误。 $f(n) = \Sigma(g(n)) \rightarrow \exists n_0, \forall n > n_0, f(n) \leq g(n)$, 而后者与之矛盾。

0.3 Prob3

证明 $lg(n!) = \Theta(nlg(n))$

$$\begin{aligned} n! &= \sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{\alpha_n} \\ lg(n!) &= lg(\sqrt{2\pi n}) + nlg(n) - nlg(e) + \alpha_n lg(e) \\ &= \Theta(lgn) + \Theta(nlgn) + \Theta(n) + \Theta\left(\frac{1}{n}\right) \\ &= \Theta(nlgn) \end{aligned}$$

证明 $n! = \omega(2^n)$

令 $n_0 = 4$, 则 $\forall n \geq n_0$, 有

$$n! = n * (n-1) * (n-2) * \dots * 4 * 3 * 2 * 1 \geq 2 * 2 * 2 * \dots * 2 * 2 * 2 * 2$$

故 $n! = \omega(2^n)$

证明 $n! = o(n^n)$

令 $n_0 = 2$, 则 $\forall n \geq n_0$, 有

$$n! = n * (n-1) * (n-2) * \dots * 2 * 1 \leq n * n * n * \dots * n * n$$

故 $n! = o(2^n)$

0.4 Prob4

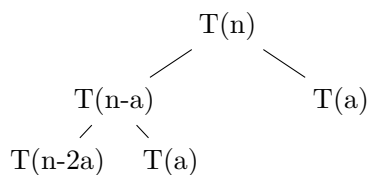
证明 假设 $T(n) = O(\lg n)$

不妨设 $T(n) = O(\lg n)$, 则对于 $n_0 = 2$, 有 $2\lg 2 \geq T(2) = T(1) + 1 = 2$

又 $\forall n > n_0$, 若已知 $T(n) \leq 2\lg n$, 有 $T(n+1) = T(\frac{n+1}{2}) + 1 \leq 2\lg(\frac{n+1}{2}) + 1 = 2\lg(n+1) - 2\lg 2 + 1 = 2\lg(n+1) - 1 \leq 2\lg(n+1)$

故 $T(n) = O(\lg n)$

0.5 Prob5



树的深度是 $\lfloor \frac{n}{a} \rfloor$, 第 h 层的值为 $c(n-ha)$ 。故可求得

$$\begin{aligned}
 O(T(n)) &= \sum_{h=0}^{\lfloor \frac{n}{a} \rfloor} c(n-ha) \\
 &= c \sum_{h=0}^{\lfloor \frac{n}{a} \rfloor} (n-ha) \\
 &= c \sum_{h=0}^{\lfloor \frac{n}{a} \rfloor} n - c \sum_{h=0}^{\lfloor \frac{n}{a} \rfloor} ha \\
 &= O(n^2)
 \end{aligned}$$

0.6 Prob6

a $T(n) = \sqrt{n} \lg n$

b $T(n) = n^2$

0.7 Prob7

可以, 由于对于递归式有 $n^{\log_2 4} < n^2 \lg n$, 故其渐进界 $O(T(n)) = n^2 \lg n$