



算法设计与分析

作业（二）

姓 名	熊恪峥
学 号	22920202204622
日 期	2022年2月28日
学 院	信息学院
课程名称	算法设计与分析

作业（二）

目录

1	题2.2	3
2	题2.9	3
3	题3.1	4
4	题3.2	5
5	题3.4	5
6	题3.6	6
7	题3.8	7

1 题2.2

由max的定义

$$\begin{aligned} f(x) &\leq \max(f(x), g(x)) \\ g(x) &\leq \max(f(x), g(x)) \end{aligned}$$

则

$$f(x) + g(x) \leq 2 \max(f(x), g(x))$$

则有

$$\max(f(x), g(x)) \geq \frac{f(x) + g(x)}{2}$$

即

$$\max(f(x), g(x)) = \Omega(f(x), g(x)) \quad (1)$$

由非负性

$$\max(f(x), g(x)) \leq f(x) + g(x)$$

即

$$\max(f(x), g(x)) = O(f(x) + g(x)) \quad (2)$$

由(1)和(2)可得

$$\max(f(x), g(x)) = \Theta(f(x) + g(x))$$

2 题2.9

必要性：由 $f(n) = \Theta(g(n))$ 得

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = k, 0 < k < \infty$$

则由

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq 0$$

得 $f(n) = \Omega(g(n))$ ，由

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq \infty$$

得 $f(n) = O(g(n))$

充分性: 由 $f(n) = O(g(n))$ 且 $f(n) = \Omega(g(n))$ 得

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq \infty$$

且

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \neq 0$$

则 $\exists k > 0$ 且 $k < \infty$ 使

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = k$$

则 $f(n) = \Theta(g(n))$

3 题3.1

正好雇佣一次, 则第一位面试的必须是最好的面试者。则

$$P(\text{排在第1位的面试者是最好的面试者}) = \frac{1}{n}$$

正好雇佣两次, 则已知排在第一位的人一定会被雇佣, 最好的面试者一定会被雇佣。因此, 第一位不能是最好的的面试者, 否则只能被雇佣一次。则设事件

E_i : 第一个来面试的人的排名是 i

其中 i 满足

$$i \leq n-1$$

则

$$P(E_i) = \frac{1}{n}$$

若第一个人的排名是 i , 只雇佣两个人要求第 $2, 3, \dots, j-1$ 个面试者排名都不如第一个面试者, 即最好的人必须在排名是 $i+1, i+2, \dots, n-1, n$ 的人中第一个面试。设最好的人面试的次序是 j , 则事件

F : 第 $2, 3, \dots, j-1$ 个面试者排名都不如第一个面试者

则

$$\begin{aligned} & \overbrace{i+1, i+2, \dots, n-1, n}^{\text{共 } n-i \text{ 个人}} \\ & \Rightarrow P(F|E_i) = \frac{1}{n-i} \end{aligned}$$

注意到 E_1, \dots, E_{n-1} 是独立事件，则由全概率公式

$$\begin{aligned} P(\text{恰好有两个人被雇佣}) &= \sum_{i=1}^{n-1} P(F|E_i) \times P(E_i) \\ &= \sum_{i=1}^{n-1} \frac{1}{n} \cdot \frac{1}{n-i} \\ &= \frac{1}{n} \sum_{i=1}^{n-1} \frac{1}{i} \\ &= O\left(\frac{\log n}{n}\right) \end{aligned}$$

正好雇佣 n 次，则所有候选人按排名单调递增进行面试。总共有 $n!$ 种排列，则概率为

$$P(\text{恰好有}n\text{个人被雇佣}) = \frac{1}{n!}$$

4 题3.2

*FindMax*算法如算法1

算法 1 查找最大值，返回下标

```

1: procedure FINDMAX( $A$ )
2:    $max \leftarrow 1$ 
3:   for  $j \leftarrow 2$  to  $n$  do
4:     if  $A[j] > A[max]$  then
5:        $max \leftarrow j$ 
   return  $max$ 

```

则 max 在位置 k 被赋予最大值时的概率为

$$P = \frac{1}{n}$$

如果 $max = A[k]$ 则第3行的比较次数为 $n - 1$ 次，则平均的比较次数为

$$\begin{aligned} T(n) &= \sum_{k=2}^n \frac{1}{n} \cdot k \\ &= \frac{1}{n} \cdot \frac{(2+n)(n-1)}{2} \\ &= \Theta(n) \end{aligned}$$

5 题3.4

设运算 i 的开销为 c_i

$$c_i = \begin{cases} i & i \text{ 为 } 2 \text{ 的整数幂} \\ 1 & \text{其它} \end{cases}$$

则总开销为

$$\begin{aligned}
 C &= \sum_{i=1}^n c_i \\
 &= \sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i + (n - \lfloor \log_2 n \rfloor) \\
 &= 2(n-1) + (n - \lfloor \log_2 n \rfloor)
 \end{aligned}$$

则平均一次运算的开销为

$$\bar{c}_i = \frac{C}{n} = 3 + \frac{1}{n} + \frac{\lfloor \log_2 n \rfloor}{n}$$

当 n 较大时

$$\begin{aligned}
 \bar{c}_i &= \lim_{n \rightarrow \infty} \frac{C}{n} = 3 + \lim_{n \rightarrow \infty} \left(\frac{1}{n} + \frac{\lfloor \log_2 n \rfloor}{n} \right) \\
 &= 3
 \end{aligned}$$

6 题3.6

已知实际代价 c_i 为

$$c_i = \begin{cases} i & i \text{ 为 2 的整数幂} \\ 1 & \text{其它} \end{cases}$$

对每一个操作收费\$3 ($\hat{c}_i = 3$)，并且

- 当 i 为2的整数幂，使用存款支付 i
- 当 i 不是2的整数幂，支付费用\$1，并且增减存款\$2

可以得到表1

表格 1: 每次操作的代价与存款

i	代价	存款
1	1	2
2	2	3
3	1	5
4	4	4
5	1	6
...

由表1可知 $\hat{c}_i = 3$ 满足

$$C = 3n = \sum_{i=0}^n \hat{c}_i \geq \sum_{i=0}^n c_i$$

则平均一次的开销为

$$\overline{c_i} = \frac{C}{n} = 3$$

7 题3.8

已知实际代价 c_i 为

$$c_i = \begin{cases} i & i \text{ 为 } 2 \text{ 的整数幂} \\ 1 & \text{其它} \end{cases}$$

根据势能法，有

$$\begin{aligned} \hat{c}_i &= c_i + D_i - D_{i-1} \\ &= \begin{cases} i + D_i - D_{i-1} & i \text{ 为 } 2 \text{ 的整数幂} \\ 1 + D_i - D_{i-1} & i \text{ 不是 } 2 \text{ 的整数幂} \end{cases} \end{aligned}$$

令 $i = 2^j + k$ ，定义势函数 $D_i = 2k$

$$D_i - D_{i-1} = \begin{cases} 0 - 2(2^j - 2^{j-1} - 1) = -i + 2 & i \text{ 为 } 2 \text{ 的整数幂} \\ 2k - 2(k-1) = 2 & i \text{ 不是 } 2 \text{ 的整数幂} \end{cases}$$

则

$$\begin{aligned} \hat{c}_i &= c_i + D_i - D_{i-1} \\ &= \begin{cases} i - i + 2 = 2 & i \text{ 为 } 2 \text{ 的整数幂} \\ 1 + 2 = 3 & i \text{ 不是 } 2 \text{ 的整数幂} \end{cases} \end{aligned}$$

求和可得

$$\begin{aligned} C = \sum_{i=0}^n \hat{c}_i &= \sum_{j=0}^{\lfloor \log_2 n \rfloor} 2 + 3(n - \lfloor \log_2 n \rfloor) \\ &= 2\lfloor \log_2 n \rfloor + 3(n - \lfloor \log_2 n \rfloor) \end{aligned}$$

平均一次的代价

$$\begin{aligned} \overline{c_i} &= \frac{C}{n} = \frac{2\lfloor \log_2 n \rfloor + 3(n - \lfloor \log_2 n \rfloor)}{n} \\ &\leq \lim_{n \rightarrow \infty} 3 + \frac{2\lfloor \log_2 n \rfloor}{n} + \frac{\lfloor \log_2 n \rfloor}{n} \\ &= 3 + 0 + 0 = 3 \end{aligned}$$