Recurrence

Heshan Du
University of Nottingham Ningbo China

March 2025

Reading

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*, Third Edition, 2009.
  - Chapter 4.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*, Fourth Edition, 2022.
  - Chapter 4.

Exercise 1

Solve the following recurrences. Note that $T(1)$ is assumed to be 1.

1. $T(n) = T(n-1)$
2. $T(n) = T(n-1) + 1$
3. $T(n) = T(n-1) + n$
4. $T(n) = 2T(n-1)$

Exercise 1: Hints

Use the substitution method.

1. Use the recurrence repeatedly, $T(n) = T(n-1) = T(n-2) = \cdots = T(1) = 1$. We guess $T(n) = 1$.
2. We guess $T(n) = n$.
3. We guess $T(n) = n(n+1)/2$.
4. We guess $T(n) = 2^{n-1}$.

Use mathematical induction to prove each guess.

Exercise 2

Solve the following recurrences.

1. $T(n) = 2T(n/2)$
2. $T(n) = 2T(n/4)$
3. $T(n) = 2T(n/2) + 1$
4. $T(n) = 2T(n/2) + n^2$

Exercise 2: Hints

1. $T(n) = \Theta(n)$
2. $T(n) = \Theta(\sqrt{n})$
3. $T(n) = \Theta(n)$
4. $T(n) = \Theta(n^2)$

Exercise 3

Solve the following recurrences.

1. $T(n) = 2T(n/2) + n + 1$
2. $T(n) = 2T(n/2) + n \log n$
3. $T(n) = 2T(n/2) + n(\log n)^2$
4. $T(n) = 4T(n/2) + n \log n$

Exercise 3: Hints

1. $T(n) = \Theta(n \log n)$
2. $T(n) = \Theta(n(\log n)^2)$
3. $T(n) = \Theta(n(\log n)^3)$
4. $T(n) = \Theta(n^2)$

Exercise 4

Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$.

Exercise 4: Solution

Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$.

Answer:

We guess that for any integer m less than n, $T(m) \leq cm^2$ holds for some constant $c > 0$. Then we have

$$\begin{aligned} T(n) &= T(n-1) + n \\ &\leq c(n-1)^2 + n \\ &= c(n^2 - 2n + 1) + n \\ &= cn^2 - 2cn + c + n \\ &= cn^2 + c(1-2n) + n. \end{aligned}$$

The last quantity is less than or equal to cn^2 if $c(1-2n) + n \leq 0$, this is, $c \geq n/(2n-1)$. This last condition holds for all $n \geq 1$ and $c \geq 1$. For the boundary condition, we set $T(1) = 1$, and so $T(1) = 1 \leq c \times 1^2$. Thus, we can choose $n_0 = 1$ and $c = 1$.

Exercise 5

Using the master method, we can show that the solution to the recurrence $T(n) = 4T(n/3) + n$ is $T(n) = \Theta(n^{\log_3 4})$. Show that a substitution proof with the assumption $T(n) \leq cn^{\log_3 4}$ fails. Then show how to substract off a lower-order term to make a substitution proof work.

Exercise 5: Solution

If we were to try a straight substitution proof, assuming that $T(n) \leq cn^{\log_3 4}$, we would get stuck:

$$\begin{aligned} T(n) &\leq 4(c(n/3)^{\log_3 4}) + n \\ &= 4c(\frac{n^{\log_3 4}}{4}) + n \\ &= cn^{\log_3 4} + n \end{aligned}$$

which is greater than $cn^{\log_3 4}$. Instead, we subtract off a lower-order term and assume that $T(n) \leq cn^{\log_3 4} - dn$. Now we have

$$\begin{aligned} T(n) &\leq 4(c(n/3)^{\log_3 4} - dn/3) + n \\ &= 4(\frac{cn^{\log_3 4}}{4} - \frac{dn}{3}) + n \\ &= cn^{\log_3 4} - \frac{4}{3}dn + n \end{aligned}$$

which is less than or equal to $cn^{\log_3 4} - dn$ if $d \geq 3$.

Exercise 6

Professor Caesar wishes to develop a matrix-multiplication algorithm that is asymptotically faster than Strassen's algorithm, whose running time is in $\Theta(n^{\log 7})$. His algorithm will use the divide-and-conquer method, dividing each matrix into pieces of size $n/4 \times n/4$, and the divide and combine steps together will take $\Theta(n^2)$ time. He needs to determine how many subproblems his algorithm has to create in order to beat Strassen's algorithm. If his algorithm creates a subproblems, then the recurrence for the running time $T(n)$ becomes $T(n) = aT(n/4) + \Theta(n^2)$. What is the largest integer value of a for which Professor Caesar's algorithm would be asymptotically faster than Strassen's algorithm?

Exercise 6: Hint

By solving $n^{\log_4 a} < n^{\log 7}$, we get $a < 49$. Hence the largest value of a is 48.

第一步：使用 Master Theorem 主定理

主定理中，设：

- $a = ?$
- $b = 4$
- $f(n) = \Theta(n^2)$
- 比较 $f(n)$ 与 $n^{\log_4 a}$

使用 **Case 1** 条件（即 $f(n) = O(n^{\log_b a - \varepsilon})$），得：

我们要的是：
$n^{\log_4 a} < n^{\log_2 7} \Rightarrow \log_4 a < \log_2 7$

将对数换底：

$$\log_4 a = \frac{\log_2 a}{\log_2 4} = \frac{\log_2 a}{2}$$

所以：

$$\frac{\log_2 a}{2} < \log_2 7 \Rightarrow \log_2 a < 2 \log_2 7 = \log_2 49$$

对两边取指数：

$$a < 49$$