Side Talk: More on Recurrences

Yufei Tao

ITEE
University of Queensland

In the class, we have started to analyze the running time of algorithms using recurrence functions. This tutorial aims at enhancing your familiarity of such functions, strengthening your ability of solving such functions, and introducing the famous "master theorem".

Exercise 1

Let $f(n)$ be a function that returns a positive value for every integer $n > 0$. We know:

$$\begin{aligned} f(1) &\leq c_1 \\ f(n) &\leq c_2 + f(n-1) \qquad (\text{for } n \geq 2) \end{aligned}$$

where c_1, c_2 are two positive constants. Prove: $f(n) = O(n)$.

Exercise 1

Proof:

$$
\begin{aligned}
f(n) &\leq c_2 + f(n-1) \\
&\leq 2c_2 + f(n-2) \\
&\leq 3c_2 + f(n-3) \\
&\ldots \\
&\leq (n-1)c_2 + f(1) \\
&\leq (n-1)c_2 + c_1 = O(n).
\end{aligned}
$$

$\square$

Exercise 2

Let $f(n)$ be a function that returns a positive value for every integer $n > 0$. We know:

$$\begin{aligned} f(1) &\leq c_1 \\ f(n) &\leq c_2\sqrt{n} + f(n-1) \qquad (\text{for } n \geq 2) \end{aligned}$$

where c_1, c_2 are two positive constants. Prove: $f(n) = O(n\sqrt{n})$.

Exercise 2

Proof:

$$
\begin{aligned}
f(n) &\leq c_2\sqrt{n} + f(n-1) \\
&\leq 2c_2\sqrt{n} + f(n-2) \\
&\leq 3c_2\sqrt{n} + f(n-3) \\
&\ldots \\
&\leq (n-1)c_2\sqrt{n} + f(1) \\
&\leq (n-1)c_2\sqrt{n} + c_1 = O(n\sqrt{n}).
\end{aligned}
$$

$\square$

Exercise 3

Let $f(n)$ be a function that returns a positive value for every integer $n > 0$. We know that when n is an even number:

$$\begin{aligned} f(1) &\leq c_1 \\ f(n) &\leq c_2 n + f(n/2) \qquad (\text{for } n \geq 2) \end{aligned}$$

where c_1, c_2 are two positive constants. Prove: $f(n) = O(n)$ when n is a power of 2.

Exercise 3

Proof:

$$
\begin{aligned}
f(n) &\leq c_2 n + f(n/2) \\
&\leq c_2 n + c_2 n/2 + f(n/2^2) \\
&\leq c_2 n + c_2 n/2 + c_2 n/2^2 + f(n/2^3) \\
&\ldots \\
&\leq c_2(n + n/2 + n/2^2 + \ldots + n/2^{h-1}) + f(n/2^h) \qquad (h = \log_2 n) \\
&\leq 2c_2 n + f(1) \qquad \text{(See the note on the next slide)} \\
&\leq 2c_2 n + c_1 = O(n).
\end{aligned}
$$

□

In the proof of the previous slide, we used the fact:

$$n + n/2 + n/2^2 + ... + n/2^{h-1} \quad \leq \quad 2n.$$

This can be generalized to the following (very useful) conclusion:

Theorem 1.

Let c be any constant strictly larger than 1. Then:

$$n + \frac{n}{c} + \frac{n}{c^2} + ... + \frac{n}{c^h} = O(n)$$

for any integer $h \geq 1$.

A proof of the theorem is provided in the next slide.

Proof of Theorem 1: Let

$$
\begin{aligned}
f(n) &= n + \frac{n}{c} + \frac{n}{c^2} + \ldots + \frac{n}{c^h} && (1)\\
\Rightarrow \frac{f(n)}{c} &= \frac{n}{c} + \frac{n}{c^2} + \frac{n}{c^3} + \ldots + \frac{n}{c^{h+1}} && (2)
\end{aligned}
$$

Subtracting (2) from (1) gives:

$$
\begin{aligned}
\left(1 - \frac{1}{c}\right) f(n) &= n - \frac{n}{c^{h+1}}\\
\Rightarrow f(n) &= \frac{n\left(1 - \frac{1}{c^{h+1}}\right)}{1 - \frac{1}{c}} \quad \text{(applying } c \neq 1\text{)}\\
&< \frac{n}{1 - \frac{1}{c}} \quad \text{(applying } c > 1\text{)}\\
&= O(n). && (3)
\end{aligned}
$$

□

Next, we will look at an exercise which may seem fairly difficult. This will provide an excellent opportunity to introduce a super-powerful theorem—the master theorem.

Exercise 4

Let $f(n)$ be a function that returns a positive value for every integer $n > 0$. We know:

$$\begin{aligned} f(1) &\leq c_1 \\ f(n) &\leq c_2\sqrt{n} + 2f(\lceil n/4 \rceil) \qquad (\text{for } n \geq 2) \end{aligned}$$

where c_1, c_2 are two positive constants. Prove: $f(n) = O(\sqrt{n}\log n)$.

Recall that $\lceil x \rceil$ (i.e., the ceiling) returns the smallest integer that is at least x. For example, $\lceil 3.5 \rceil = 4$ and $\lceil 3 \rceil = 3$.

The master theorem, introduced in the next slide, will solve the exercise immediately.

The Master Theorem

Theorem 2.

Let $f(n)$ be a function that returns a positive value for every integer $n > 0$. We know:

$$\begin{aligned} f(1) &\leq c_1 \\ f(n) &\leq \alpha \cdot f(\lceil n/\beta \rceil) + c_2 \cdot n^\gamma \qquad (\text{for } n \geq 2) \end{aligned}$$

where α, β, γ, c_1, and c_2 are positive constants. Then:

- If $\log_\beta \alpha < \gamma$, then $f(n) = O(n^\gamma)$.
- If $\log_\beta \alpha = \gamma$, then $f(n) = O(n^\gamma \log n)$.
- If $\log_\beta \alpha > \gamma$, then $f(n) = O(n^{\log_\beta \alpha})$.

The proof the theorem is a bit tedious and omitted from this course.

Exercise 4

Proof: In the given recurrence, $\alpha = 2, \beta = 4$, and $\gamma = 1/2$, where α, β, as γ are as defined in the master theorem. Since

$$\log_\beta \alpha = \log_4 2 = 1/2 = \gamma$$

the master theorem indicates that $f(n) = O(n^\gamma \log n) = O(\sqrt{n} \log n)$. This completes the proof. $\square$

Exercise 5

Let $f(n)$ be a function that returns a positive value for every integer $n > 0$. We know:

$$\begin{aligned} f(1) &\leq c_1 \\ f(n) &\leq c_2\sqrt{n} + 2f(\lceil n/5 \rceil) \qquad (\text{for } n \geq 2) \end{aligned}$$

where c_1, c_2 are two positive constants. Prove: $f(n) = O(\sqrt{n})$.

Exercise 5

Proof: In the given recurrence, $\alpha = 2, \beta = 5$, and $\gamma = 1/2$, where α, β, as γ are as defined in the master theorem. Since

$$\log_\beta \alpha = \log_5 2 < 1/2 = \gamma$$

the master theorem indicates that $f(n) = O(n^\gamma) = O(\sqrt{n})$. This completes the proof. $\square$

Here is one more exercise for you (which should be straightforward now)—apply the master theorem to solve Exercise 3.