Questions

[If you are studying for the final exam and only have a few hours to spend on these particular exercises, then I suggest concentrating on just a few of them, say Q7-Q9 and Q13-15.]

1. Suppose you have three $n \times n$ arrays, call them `a[][]`, `b[][]`, and `c[][]`. Consider the following.

```
for i = 1 to n
   for j = 1 to n{
     c[i][j] = 0
     for k = 1 to n
        c[i][j] += a[i][k] * b[k][j];
   }
```

   (Those you familiar with linear algebra will recognize this as matrix multiplication.)

   Give a tight big O bound on this algorithm as a function of n.

2. True or false? Prove it.
   (a) $n!$ is $O((n+2)!)$.
   (b) $(n+2)!$ is $O(n!)$.
   (c) 9^n is $O(12^n)$.
   (d) 12^n is $O(9^n)$.

3. Let $t(n) = \sum_{i=0}^{n} 3^i$. Show that $t(n)$ is $O(3^n)$.

4. (a) Use mathematical induction to prove that $Fib(n)$ is $O(\ (\frac{7}{4})^n\ )$.
   (b) Use mathematical induction to prove that $Fib(n) \in \Omega((\frac{3}{2})^n)$.

5. Show 2^n is $O(n!)$.

6. Let $t(n) = n \log n$. Prove that $t(n)$ is $\Omega(\log(n!))$.

7. Let $t(n) = \frac{n^2}{2} + 3\log n - 40.$. Prove that $t(n)$ is $\Omega(n^2)$.

8. Let $t(n) = \frac{1}{5}\log(n-8)$. Show that $t(n)$ is $\Omega(\log(n))$

9. Let $t(n) = (n+8)^{1.3} + 3n + 5$. Prove that $t(n)$ is $O(n^{1.3})$.

10. Let $t(n) = \sqrt{31n + 12n\log n + 57}$. Prove that $O(\sqrt{n}\log n)$.

 ©Michael Langer

11. If $f(n)$ is $O(g(n))$, can we conclude that $2^{f(n)}$ is $O(2^{g(n)})$?

12. Is $t(n) = \frac{1}{n}$ in $\Omega(1)$?

13. Let $t(n) = 5n^2 + 3n + 4$.

    (a) Use a limit argument to show that $t(n)$ is $O(n^2)$.

    (b) Find constants c, n_0 that satisfy the definition of big O for this example.

14. Give a tight big O bound on

$$t(n) \ = \ \sqrt{n^2 + 100n} - n.$$

15. What are the $O()$ and $\Omega()$ relationships between $t(n) = n^a$ and $g(n) = n^b$, where $0 < a < b$?

16. What is the big O and big Omega relationship between $t(n) = \log_a n$ and $g(n) = \log_b n$, where $0 < a < b$?

    Hint:

$$\log_a n = \log_a b \ * \ \log_b n$$

©Michael Langer

Answers

1. The algorithm is $O(n^3)$. Why? For each value of i, we run the two inner loops (j and k). There are n values of i, so the number of steps is n times the number of steps in the two inner loops. The two inner loops take n^2 steps (by similar reasoning, namely for each value of j, we run through all n values of k). Thus, the number of steps is $O(n * n^2) = O(n^3)$.

2. (a) (True) Applying the formal definition, we want to know if

$$n! < c(n+2)(n+1) \cdot n!$$

for n sufficiently large. Dividing by $n!$ gives

$$1 < c(n+2)(n+1).$$

So let $c = 1$ and $n_0 = 1$.

(b) (False) Here we need to find a $c, n_0 > 0$ such that

$$(n+2)(n+1) \cdot n! \ < \ c(n!)$$

for all $n > n_0$. Choose any c, n_0. Then, dividing by $n!$, we would now need to show that $(n+1)(n+2) < c$ for all $n \geq n_0$. But this is clearly false, since the left side grows without bound as n grows. Thus, $(n+2)!$ is not $O(n!)$.

(c) (True) Since $9 < 12$, it follows that $9^n < 12^n$ and so $c = 1$ and $n_0 = 1$ does the job.

(d) (False) We want to show there exists $c, n_0 > 0$ such that $12^n < c9^n$ for all $n \geq n_0$. But

$$12^n < c9^n \iff (\frac{12}{9})^n < c$$

But this inequality cannot be true for all $n \geq n_0$, since the left side grows without bound. Thus, 12^n cannot be $O(9^n)$.

3. Recall the formula for a geometric series

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1} - 1}{a - 1}.$$

Then,

$$\sum_{i=1}^{n} 3^i = \frac{3^{n+1} - 1}{3 - 1} = \frac{3}{2}(3^n - \frac{1}{3})$$

which is $O(3^n)$, i.e. take $c = \frac{3}{2}$ and $n_0 = 1$.

4. (a) We need to find an n_0 and c such that, for all $n \geq n_0$, $F(n) < c(\frac{7}{4})^n$.

Try $c = 1$. The base case is trivial since $F(0) = 0 < (\frac{7}{4})^0$ and $F(1) = 1 < \frac{7}{4}$. So let's hypothesize that $F(n) < (\frac{7}{4})^n$ for all n up to some $k \geq 1$ and see if it follows for $n = k+1$.

$$\begin{aligned} F(k+1) &= F(k) + F(k-1) \\ &< (\frac{7}{4})^k \ + \ (\frac{7}{4})^{k-1} \ \text{ by the induction hypothesis,} \\ &= (\frac{7}{4} \ + \ 1)(\frac{7}{4})^{k-1} \end{aligned}$$

But it is easy to verify that $\frac{7}{4} \ + \ 1 < (\frac{7}{4})^2$ and so (from the induction hypothesis) we get

$$\begin{aligned} F(k+1) &< (\frac{7}{4})^2 \ (\frac{7}{4})^{k-1} \\ &= (\frac{7}{4})^{k+1}. \end{aligned}$$

This proves the induction step, and so we are done.

(b) We need to find an n_0 and c such that $F(n) > c(\frac{3}{2})^n$ for all $n \geq n_0$.
Let's first establish a base case. We can't have a base case for $n = 0$ since $F(0) = 0$ and so it will be impossible for $F(0) > c(\frac{3}{2})^0$ for $c > 0$. Instead, we try to find a c and use the base case(s) $n = 1, 2$. If we let $c = (\frac{2}{3})^2$, then indeed we have $F(n) > c(\frac{3}{2})^n$ for $n = 1, 2$. So let's try using that c and proving the induction step.
We assume the induction hypothesis, namely we assume that $F(n) > c(\frac{3}{2})^n$ for $n = k-1, k$. We want to show it follows that $F(k+1) > c(\frac{3}{2})^{k+1}$.

$$\begin{aligned} F(k+1) &= F(k) + F(k-1) \\ &> c(\frac{3}{2})^k \ + \ c\ (\frac{3}{2})^{k-1} \ \text{ by induction hypothesis} \\ &= c\ (\frac{3}{2} + 1)\ (\frac{3}{2})^{k-1} \\ &> c\ (\frac{3}{2})^2 (\frac{3}{2})^{k-1}, \ \text{since } \frac{5}{2} > \frac{9}{4} \\ &= c\ (\frac{3}{2})^{k+1} \end{aligned}$$

Thus, both the base case and induction step are proved and so we are done.

5. We want to show that there exist two constants $c > 0$ and $n_0 > 0$ such that, for all $n \geq n_0$,

$$2^n \ \leq \ c\ n!$$

or, equivalently,

$$\frac{2}{n} \cdot \frac{2}{n-1} \cdot \frac{2}{n-2} \cdots \frac{2}{4} \cdot \frac{2}{3} \cdot \frac{2}{2} \cdot \frac{2}{1} \leq c.$$

On the left side, the numerator and denominator have n terms each. We pair them up and note that numerator terms are all less than or equal to their corresponding denominator terms, except for the last pair $(\frac{2}{1})$. We take the last pair to the other side,

$$\frac{2}{n} \cdot \frac{2}{n-1} \cdot \frac{2}{n-2} \cdots \frac{2}{4}\frac{2}{3} \cdot \frac{2}{2} \leq \frac{c}{2}.$$

 ©Michael Langer

The terms on the left side are for $n \geq 2$. If $n = 1$, then the left side is 1.

So, if we let $c = 2$ and $n_0 = 1$, then this inequality indeed is true for all $n \geq n_0$ since the right side is 1 and the left side is a product of terms that are each less than or equal to 1.

6. We need to show there exist two positive constants c, n_0 such that, for all $n \geq n_0$,

$$n \log n > c \log(n!).$$

Try $c = 1$. Since $n \log n = \log(n^n)$, and since $\log x$ is monotonically increasing, it is enough for us to show that there exist n_0 such that, for all $n \geq n_0$,

$$n^n > n!$$

But it is easy to see that $\frac{n^n}{n!} > 1$ since both numerator and denominator have n terms each, and if we take corresponding terms, we notice that the ratio is greater than or equal to 1 for each. Thus, the product of the ratios is greater than or equal to 1.

7. Here are two ways to do it. The first way:

$$\begin{aligned} t(n) &= \frac{n^2}{2} + 3 \log n - 40 \\ &\geq \frac{n^2}{2} - 40 \text{ for } n \geq 1 \end{aligned}$$

Since we are looking for a lower bound, let's try a constant $c < \frac{1}{2}$, specifically take $c = \frac{1}{4}$. We want to find an n_0 such that, for all $n \geq n_0$,

$$\frac{n^2}{2} - 40 > \frac{n^2}{4}$$

or equivalently

$$\frac{n^2}{4} > 40$$

We see $n_0 = 13$ does the job, since $13^2 = 169 > 160 = 4 * 40$.

The second way to do it is to guess $c = \frac{1}{2}$ and then find an n_0 such that $3 \log n - 40 > 0$ for all $n > n_0$. Choosing $n_0 = 2^{\frac{40}{3}}$ does the job.

8. We are looking for a lower bound so let's try some constant $c < \frac{1}{5}$. Let's try $c = \frac{1}{10}$.

$$\begin{aligned} & \frac{1}{5} \log(n-8) > \frac{1}{10} \log n \\ \iff & \log(n-8) > \frac{1}{2} \log n \\ \iff & \log(n-8) > \log \sqrt{n} \\ \iff & n - 8 > \sqrt{n} \end{aligned}$$

 ©Michael Langer

But the last inequality is true if n is sufficiently large, since n grows faster than $\sqrt{n}$. We still need to choose an n_0. The inequality holds for $n_0 = 16$ since $8 > 4$. Moreover, dividing both sides by $\sqrt{n}$ gives

$$\sqrt{n} > 1 + \frac{8}{\sqrt{n}}$$

which holds for all $n > 16$ since the left side is increasing and the right side is decreasing. So, $n_0 = 16$ does the job (and $c = \frac{1}{10}$).

9. We need to show there exists two positive constants c, n_0 such that, for all $n \geq n_0$,

$$(n+8)^{1.3} + 3n + 5 < cn^{1.3}.$$

$$\begin{aligned}
(n+8)^{1.3} + 3n + 5 &< (2n)^{1.3} + 3n + 5, \quad \text{if } n \geq 8 \\
&< 4n^{1.3} + 3n^{1.3} + 5n^{1.3}, \quad \text{since } 2^{1.3} < 2^2 = 4 \\
&= 12n^{1.3}
\end{aligned}$$

So, take $n_0 = 8$ and $c = 12$.

10. We want to show there exists a $c > 0$ and $n_0 \geq 1$ such that, for all $n \geq n_0$,

$$\sqrt{31n + 12n\log n + 57} < c\sqrt{n}\log n.$$

But

$$\begin{aligned}
\sqrt{31n + 12n\log n + 57} &< \sqrt{31n\log n + 12n\log n + 57n\log n}, \text{ when } \ n > 2 \\
&= \sqrt{100n\log n} \\
&= 10\ \sqrt{n}\ \sqrt{\log n} \\
&< 10\sqrt{n}\log n, \text{ when } \ n > 2
\end{aligned}$$

where the last line follows from the fact that $\sqrt{x} < x$ when $x > 1$. So, take $n_0 = 3$ and $c = 10$.

11. No. Take $f(n) = 2n$ and $g(n) = n$. However, 2^{2n} is 4^n which is not $O(2^n)$.

12. The definition of $\Omega()$ requires $c > 0$. However, for any such c that we choose, there will be an n_0 such that $t(n) < c$ when $n \geq n_0$, namely $n_0 = \frac{1}{c}$. The idea here is that $t(n)$ is not asymptotically bounded below by a strictly positive constant.

13. (a) When we compute the limit, we get:

$$lim_{n\to\infty} \ \frac{5n^2 + 3n + 4}{n^2} = 5$$

So, the third limit rule gives us that $t(n)$ is $\Theta(g(n))$, and thus in particular $t(n)$ is $O(g(n))$.
[ASIDE: You might be thinking you would use the first limit rule using limits which said that if $lim_{n\to\infty}\frac{t(n)}{g(n)} = 0$ then $t(n)$ is $O(g(n))$. However, that rule doesn't apply here.]

 ©Michael Langer

(b) Since the limit is 5, you might be tempted to choose $c = 5$ as your constant. However, if you plug $c = 5$ into the inequality $t(n) \le cn^2$, you see it never is true.

As an alternative, find an upper bound on $t(n)$ as follows:

$$5n^2 + 3n + 4 < 5n^2 + 3n^2 + 4n^2 = 12n^2$$

and so we can take $c = 12$ and $n_0 = 1$.

14. You might guess that $t(n)$ is $O(n)$. Let's see what happens when we compute:

$$lim_{n\to\infty} \frac{t(n)}{n} = lim_{n\to\infty} \frac{\sqrt{n^2 + 100n} - n}{n} = lim_{n\to\infty} \sqrt{1 + \frac{100}{n}} - 1 = 0.$$

Since the limit is 0, $t(n)$ is not $\Theta(n)$. But what is the tighter upper bound? In fact, $t(n)$ is $O(1)$. This is a bit tricky to prove using limits, so let's instead show it by finding an explicit constant upper bound.

$$\begin{aligned} t(n) &= \sqrt{n^2 + 100n} - n \\ &\le \sqrt{n^2 + 100n + 2500} - n \\ &= \sqrt{(n + 50)^2} - n \\ &= n + 50 - n \\ &= 50 \end{aligned}$$

So, $t(n)$ is bounded above by a constant for all n, which means $t(n)$ is $O(1)$.

15. Since $b > a$ we have that $\lim_{n\to\infty} \frac{n^a}{n^b} = \lim_{n\to\infty} \frac{1}{n^{b-a}} = 0$. Thus, n^a is $O(n^b)$ but n^a is not $\Omega(n^b)$.

16. Since

$$\log_a n = \log_a b \ * \ \log_b n$$

they differ by a constant factor only, and so they are in the same Θ class.

 ©Michael Langer