Test 1 — takehome portion

Nathan Typanski

March 22, 2015

Problem 1. Given are two functions A, B defined as $A(n) = n^{\lg c}$ and $B(n) = c^{\lg n}$ for some constant $c > 1$. Indicate and justify whether $A \in O(B)$, $A \in \Theta(B)$, and $A \in \Omega(B)$, respectively.

Solution. Since for any x and y we have that $x^{\log_b y} = y^{\log_b x}$, we may conclude that $n^{\lg c} = c^{\lg n}$, and we have that $n^{\lg c} \in O(c^{\lg n})$, $n^{\lg c} \in \Theta(c^{\lg n})$, and $n^{\lg c} \in \Omega(c^{\lg n})$.

Problem 2. Show that $\lg(N!) \in \Theta(N \lg N)$.

Solution. Gosper's version of Sterling's approximation of the factorial is

$$n! \approx n^n e^{-n} \sqrt{\left(2n + \frac{1}{3}\right)\pi}. \tag{1}$$

Using (1) to approximate $n!$ in the expression $\lg n!$,

$$\lg n! \approx \lg \left[n^n e^{-n} \sqrt{\left(2n + \frac{1}{3}\right)\pi} \right]. \tag{2}$$

Recalling that $\lg(xy) = \lg x + \lg y$,

$$\lg n! \approx \lg n^n + \lg e^{-n} + \lg \left(\left(2n + \frac{1}{3}\right)\pi \right)^{-1/2}$$

which, using the identity $\lg x^d = d \lg x$, is equivalent to

$$\lg n! \approx n \lg n - n \lg e - \frac{1}{2} \lg \left(\left(2n + \frac{1}{3}\right)\pi \right). \tag{3}$$

Collecting constants in (3),

$$\lg n! \approx n \lg n - n \lg e - \frac{1}{2} \lg \left(2n + \frac{1}{3}\right) - \frac{1}{2} \lg \pi. \tag{4}$$

$$\lg n! \approx n \lg n - n \lg e - \frac{1}{2} \lg \left(\frac{6}{3} \left(n + \frac{1}{6} \right) \right) - \frac{1}{2} \lg \pi. \tag{5}$$

$$\lg n! \approx n \lg n - n \lg e - \frac{1}{2} \lg \frac{6}{3} - \frac{1}{2} \lg \left(n + \frac{1}{6} \right) - \frac{1}{2} \lg \pi. \tag{6}$$

Ignoring constant terms, (6) simplifies to

$$\lg n! \approx n \lg n - n \lg e - \frac{1}{2} \lg \left(n + \frac{1}{6} \right). \tag{7}$$

Problem 3. What is the probability p that a randomly selected array of size n is reverse-sorted, if all elements are distinct?

A. $p = \Theta(n \lg n)$

B. $p = 1/(n!)$

C. $p = O(1)$

D. None of the above.

Solution. The answer is *B*. Assuming a random distribution with equal probability of each arrangement, there are $n!$ possible arrangements of the array, and each one has $1/n!$ probability.

Problem 4. What does it mean for an algorithm to be correct?

A. It always terminates.

B. It always runs within the proven asymptotic bounds.

C. It always produces the right answer.

D. None of the above.

Solution. The answer is *C—It always produces the right answer.*

Problem 5. An algorithm can do no better than the following summation:

$$T(n) = \sum_{j=3}^{n} 2j^2 + 11. \tag{8}$$

Represent the summation in closed form, then in asymptotic notation.

Solution.

$$T(n) = \sum_{j=3}^{n} 2j^2 + 11$$

$$T(n) = 2\sum_{j=3}^{n}(j^2) + 11$$

Recalling that

$$\sum_{k=1}^{n}(k^2) = \frac{n(n+1)(2n+1)}{6} \tag{9}$$

and

$$\sum_{j=3}^{n} j^2 = \sum_{j=1}^{n} j^2 - \sum_{j=1}^{2} j^2 \tag{10}$$

we can simplify $T(n)$ using (10) into

$$T(n) = 2\left[\sum_{j=1}^{n} j^2 - \sum_{j=1}^{2} j^2\right] + 11.$$

Using (9),

$$T(n) = 2\left[\frac{n(n+1)(2n+1)}{6} - \frac{2(2+1)\big(2(2)+1\big)}{6}\right] + 11$$

$$T(n) = 2\left[\frac{(n^2+n)(2n+1)}{6} - 5\right] + 11$$

$$T(n) = \frac{2n^3 + 3n^2 + n}{3} + 1$$

leaving us with

$$T(n) = \frac{2n^3}{3} + n^2 + \frac{n}{3} + 1$$

which is $O(n^3)$.

Problem 6. Strassen invented a divided and conquer matrix multiplication algorihtm which theoretically beats the standard multiplication algorihtm. His algorithm is described by the following recurrence. Give bounds for (solve) the following recurrence:

$$T(n) = 7T(n/2) + \Theta(n^2). \tag{11}$$

Solution. Use the master theorem, $a = 7$, $b = 2$, and $f(n) = \Theta(n^2)$. By case 1, $f(n) \in O(n^{\lg 7 - \epsilon})$ for some $\epsilon > 0$ (note that $\lg 7 \approx 2.8$) and $T(n) \in \Theta(n^{\lg 7})$.

Problem 7. CountSort and RadixSort are $O(n+k)$ and $O(nk)$ respectively. If I want to sort 1 million 16-bit integers, what is the difference in the interpretation (value) of k for these two algorithms?

Solution. CountSort is $O(n+k)$. If we are sorting 1 million 16-bit integers, then k refers to $2^{16} = 65536$. RadixSort, by contrast, is $O(nk)$. In this case, k refers to the quantity *1 million.*