

Asymptotic Notation

We begin by stating a few useful definitions.

1. If $f(n) = \Theta(g(n))$, then \exists positive constants c_1, c_2, n_0 such that $0 \leq c_1g(n) \leq f(n) \leq c_2g(n)$, for all $n \geq n_0$.
2. If $f(n) = O(g(n))$, then \exists positive constants c, n_0 such that $0 \leq f(n) \leq cg(n)$, for all $n \geq n_0$.
3. If $f(n) = \Omega(g(n))$, then \exists positive constants c, n_0 such that $0 \leq cg(n) \leq f(n)$, for all $n \geq n_0$.
4. If $f(n) = o(g(n))$, then \exists positive constants c, n_0 such that $0 \leq f(n) < cg(n)$, for all $n \geq n_0$.
5. If $f(n) = \omega(g(n))$, then \exists positive constants c, n_0 such that $0 \leq cg(n) < f(n)$, for all $n \geq n_0$.

Proof by Substitution

It is easy to complete a proof by substitution, and a bit harder to ascertain if you have found the best possible upper bound. For example, we will show that $T(n) \leq an + T(9/10n)$ is $O(n \log n)$ (which is not the best bound). In order to prove this, we will apply the substitution method, which is basically induction. We will assume that the asymptotic running time bound holds for small n , assume it is true for all $n \leq n'$, and then show that it is true for all $n > n'$. Thus, in this case, we assume $T(n) \leq cn \log n$.

$$T(n) = an + T(9/10n) \tag{1}$$

$$\leq an + c(9n/10) \log (9n/10) \tag{2}$$

$$= an + c(9n/10)(\log (9/10) + \log n) \tag{3}$$

$$\leq an + c(9n/10) \log n \tag{4}$$

$$\leq an \log n + c(9n/10) \log n \tag{5}$$

$$\leq n \log n(a + 9/10c) \tag{6}$$

But what we want is to show that $T(n) \leq cn \log n$. Thus, we want:

$$a + 9/10c \leq c \tag{7}$$

$$a \leq c/10 \tag{8}$$

$$10a \leq c \tag{9}$$

Thus, we have bounded $c...$ as long as $c \geq 10a$, then $T(n) = O(n \log n)$. But this is not the sharpest upper bound. Now, using the same methodology, we show that $T(n) = O(n)$.

$$T(n) = an + T(9/10n) \tag{10}$$

$$\leq an + c(9n/10) \tag{11}$$

$$n(a + 9/10c) \leq cn \tag{12}$$

$$a + 9/10c \leq c \tag{13}$$

$$a \leq c/10 \tag{14}$$

$$10a \leq c \tag{15}$$

Thus, again we have bound $T(n)$. By choosing $c \geq 10a$, then $T(n) = O(n)$. Note: when doing proofs by substitution, it is always important to remember the precise goal. For example, consider this **INCORRECT PROOF** that $T(n) = an + 2T(n/2)$ is $O(n)$.

$$T(n) = an + 2T(n/2) \tag{16}$$

$$\leq an + 2(cn/2) \tag{17}$$

$$\leq (a + c)n = O(n) \tag{18}$$

This is an absolutely true statement. $(a + c)n = O(n)$. However, what we needed to show was that $(a + c)n \leq cn$ for some choice of c . This will never be true. When we assume that $T(n) = O(n \log n)$, the correct proof is as follows.

$$T(n) = an + 2T(n/2) \tag{19}$$

$$\leq an + 2(cn/2 \log n/2) \tag{20}$$

$$= an + cn(\log n - \log 2) \tag{21}$$

$$= an + cn(\log n - 1) \tag{22}$$

$$an + cn(\log n - 1) \leq cn \log n \tag{23}$$

$$a + c(\log n - 1) \leq c \log n \tag{24}$$

$$a - c + c \log n \leq c \log n \tag{25}$$

$$a \leq c \tag{26}$$

Thus, this equation is true for all $c \geq a$. Thus $T(n) = an + 2T(n/2)$ is $O(n \log n)$.

Random Search

The purpose of this example is to derive some classic results in a very simple and clear way. In RANDOM-SEARCH, we always have an array of size n , and we are looking for a particular element x in that array, or an index i such that $A[i] = x$.

- a. For this analysis, let us assume that we choose and replace. In other words, we choose a number i at random from between 1 and n each time (we toss the elements into a bag and pluck them out one by one, returning them each time). If $A[i] = x$, we return, else we continue to choose and replace elements. In order to analyze the expected running time of our algorithm, let Z be a random variable such that Z equals the number of indices picked before x is found. Since we allow indices to be chosen more than once, $E[Z]$ can be calculated as follows:

$$E[Z] = \sum_{z=1}^{\infty} z \cdot Pr[Z = z] \tag{27}$$

$$E[Z] = 1 \cdot Pr[Z = 1] + 2 \cdot Pr[Z = 2] + 3 \cdot Pr[Z = 3] + \dots \tag{28}$$

$$E[Z] = 1 \cdot \frac{1}{n} + 2 \cdot \frac{n-1}{n} \cdot \frac{1}{n} + 3 \cdot \left(\frac{n-1}{n}\right)^2 \cdot \frac{1}{n} + \dots \tag{29}$$

$$E[Z] = \frac{1}{n} \left(1 + 2 \cdot \frac{n-1}{n} + 3 \cdot \left(\frac{n-1}{n}\right)^2 + \dots\right) \tag{30}$$

$$E[Z] = \frac{1}{n} \sum_{j=0}^{\infty} (j+1) \left(\frac{n-1}{n}\right)^j \tag{31}$$

We can simplify the expression for $E[Z]$ by using the following result:

$$\sum_{j=0}^{\infty} x^{j+1} = x \sum_{j=0}^{\infty} x^j \quad (32)$$

$$= \frac{x}{1-x} \quad (33)$$

$$\frac{d}{dx} \left(\sum_{j=0}^{\infty} x^{j+1} \right) = \sum_{j=0}^{\infty} (j+1)x^j \quad (34)$$

$$= \frac{1}{(1-x)^2} \quad (35)$$

Therefore, the $E[Z]$ can be written in the following way:

$$E[Z] = \frac{1}{n} \frac{1}{\left(1 - \frac{n-1}{n}\right)^2} \quad (36)$$

$$E[Z] = \frac{1}{n} \frac{1}{\left(\frac{n-(n-1)}{n}\right)^2} \quad (37)$$

$$E[Z] = \frac{1}{n} \frac{1}{\left(\frac{1}{n^2}\right)} \quad (38)$$

$$E[Z] = n \quad (39)$$

This is a bit surprising. Searching an array of size n deterministically takes $O(n)$ time of course. But now we see that searching at random takes expected time $O(n)$ as well.

- b. Now, find the expected number of indices picked when each index is inspected only once. In other words, we toss our elements into a bag, choose them 1 by 1, and discard each time if $A[i] \neq x$. Let Z be a random variable such that Z equals the number of indices picked before x is found. $E[Z]$ can be calculated as follows:

$$E[Z] = 1 \cdot Pr[Z = 1] + 2 \cdot Pr[Z = 2] + 3 \cdot Pr[Z = 3] + \dots + n \cdot Pr[Z = n] \quad (40)$$

$$E[Z] = 1 \cdot \frac{1}{n} + 2 \cdot \frac{n-1}{n} \cdot \frac{1}{n-1} + 3 \cdot \frac{n-1}{n} \cdot \frac{n-2}{n-1} \cdot \frac{1}{n-2} + \dots + n \cdot \frac{n-1}{n} \cdot \frac{n-2}{n-1} \dots \frac{1}{n-(n-1)} \quad (41)$$

$$E[Z] = 1 \cdot \frac{1}{n} + 2 \cdot \frac{1}{n} + 3 \cdot \frac{1}{n} + \dots + n \cdot \frac{1}{n} \quad (42)$$

$$E[Z] = \frac{1}{n} (1 + 2 + \dots + n) \quad (43)$$

$$E[Z] = \frac{1}{n} \cdot \frac{n(n+1)}{2} \quad (44)$$

$$E[Z] = \frac{n+1}{2} \quad (45)$$

This is to be expected. In order to find the desired element, we have to look at about half.

- c. For this analysis, we assume that x is not unique in the array. In fact, we assume that the array A contains k x 's. As in part a, we choose and replace. Again, let Z be a random variable such that Z equals the number of indices picked before x is found. Since there can now be multiple elements k in the array that equal x , we modify the expression for $E[Z]$ in the following way and use the result from

part a to simplify:

$$E[Z] = 1 \cdot Pr[Z = 1] + 2 \cdot Pr[Z = 2] + 3 \cdot Pr[Z = 3] + \dots \quad (46)$$

$$E[Z] = 1 \cdot \frac{k}{n} + 2 \cdot \frac{n-k}{n} \cdot \frac{k}{n} + 3 \cdot \left(\frac{n-k}{n}\right)^2 \cdot \frac{k}{n} + \dots \quad (47)$$

$$E[Z] = \frac{k}{n} \left(1 + 2 \cdot \frac{n-k}{n} + 3 \cdot \left(\frac{n-k}{n}\right)^2 + \dots\right) \quad (48)$$

$$E[Z] = \frac{k}{n} \sum_{j=0}^{\infty} (j+1) \left(\frac{n-k}{n}\right)^j \quad (49)$$

$$E[Z] = \frac{k}{n} \frac{1}{\left(1 - \frac{n-k}{n}\right)^2} \quad (50)$$

$$E[Z] = \frac{k}{n} \frac{1}{\left(\frac{k}{n}\right)^2} \quad (51)$$

$$E[Z] = \frac{n}{k} \quad (52)$$

Again, this is to be expected.

- d. In order to determine the expected number of indices picked until the entire array has been searched, define the following variables. Let Z be a random variable such that Z equals the number of indices picked until the entire array has been searched. Let Y_i equal the number of indices picked since the i -th index was found and until the $(i+1)$ -th index is found. Therefore,

$$Z = Y_0 + Y_1 + Y_2 + \dots + Y_{n-1} \quad (53)$$

$$E[Z] = E[Y_0 + Y_1 + Y_2 + \dots + Y_{n-1}] \quad (54)$$

$$E[Z] = E[Y_0] + E[Y_1] + E[Y_2] + \dots + E[Y_{n-1}] \quad (55)$$

We can calculate the $E[Y_i]$ directly. Remember that Y_i is the number of tries to find the $(i+1)$ -th indice after we have already found i indices:

$$E[Y_i] = 1 \cdot Pr[Y_i = 1] + 2 \cdot Pr[Y_i = 2] + 3 \cdot Pr[Y_i = 3] + \dots \quad (56)$$

$$E[Y_i] = 1 \cdot \frac{n-i}{n} + 2 \cdot \frac{i}{n} \cdot \frac{n-i}{n} + 3 \cdot \left(\frac{i}{n}\right)^2 \cdot \frac{n-i}{n} + \dots \quad (57)$$

$$E[Y_i] = \frac{n-i}{n} \left(1 + 2 \cdot \frac{i}{n} + 3 \cdot \left(\frac{i}{n}\right)^2 + \dots\right) \quad (58)$$

$$E[Y_i] = \frac{n-i}{n} \sum_{j=0}^{\infty} (j+1) \left(\frac{i}{n}\right)^j \quad (59)$$

$$E[Y_i] = \frac{n-i}{n} \frac{1}{\left(1 - \frac{i}{n}\right)^2} \quad (60)$$

$$E[Y_i] = \frac{n-i}{n} \frac{1}{\left(\frac{n-i}{n}\right)^2} \quad (61)$$

$$E[Y_i] = \frac{n}{n-i} \quad (62)$$

Now we can calculate the $E[Z]$. However, a closed form of this particular solution is not known to me at this time and so I will leave it in the unsimplified form:

$$E[Z] = \sum_{i=1}^{n-1} \frac{n}{n-i} \quad (63)$$