17.2 Big O vs. Worst Case

A short digression on asymptotics

Consider the following statements about BSTs. Which of the following are true?

1. The worst-case height of a BST is $\Theta(N)$.
2. BST height is $O(N)$.
3. BST height is $O(N^2)$.

The answer is that all three statements are true. BSTs always have a height that is linear or better, and a linear height is obviously "less than" the quadratic upper bound in the last point.

However, a more tricky question is which of the three statements is *the most informative*.

The answer here is the first statement: it gives an *exact* upper and lower bound unlike the other statements. $O(N)$ could mean linear, logarithmic, square-root, or constant, but $\Theta(N)$ can only mean linear.

For an analogy, consider the following statements about the worst-case cost of a hotel room:

1. The most expensive room is $639/night.
2. The most expensive room is less than or equal to $2000/night.

Here, we see that the first statement gives us exact information, whereas the second statement does not. In the second statement, the most expensive room could be $2000, $10, or anywhere in between.

However, *both are statements about the worst case*. Applying this to asymptotic notation, this means that we can refer to the worst case with Θ, O, or even Ω. **Big O is not the same as the worst case!**

Using Big O

If Θ is always more informative than O, then why do we bother using Big O notation at all? There are several reasons:

- We can make broader statements. For example, saying "binary search is $O(\log N)$ is correct, but saying "binary search tree is $\Theta(logN)$" would not be correct, since it can be constant in certain scenarios.
- Sometimes, it is not possible or extremely difficult to determine the exact runtime. In such cases, we would still like to provide a generalized upper bound.

Previous
17.1 BST Performance

Next
17.3 B-Tree Operations