Max Array Problem Solution
Brian Loughran
Data Structures
Johns Hopkins

The cost of the max array problem is O(n).

Imagine an array that looks like such:

$$[1, 4, 8, 2, 7, 4, 0, 2]$$

This can be split into two arrays like such:

$$[1, 4, 8, 2] \quad [7, 4, 0, 2]$$

And further split into four arrays like so:

$$[1, 4] \quad [8, 2] \quad [7, 4] \quad [0, 2]$$

Each split is an operation of O(1). You will need O(n) splits to get the array in this form, so that is O(n) time.

This is beginning to look like a standard sports playoff bracket to me, where the winner of the bracket is the largest value. While I chose a nice number for demonstration purposes, you can still set up a bracket like this for any size n. For example, in the NFL, there are 12 teams that make the playoffs. The way that this works is that 4 teams are given bye weeks. So in our example, some indices of the array may be given a “bye week”.

Regardless of the size of the array, it is clear that before we determine the largest number, each other number must be “eliminated” from the array (save for the largest value). That comes out to O(n) operations of O(1) time, which gives O(n).

O(n) + O(n) = O(n).

Therefore we can solve the max array problem in O(n) time.