

Kevin Sliker
October 16, 2015
Part 2

Amortized Cost Exercise

1. Performing 16 push operations on an array of capacity 8, when doubling array capacity, would be a total cost of $8 + 9 + 7 = 24$. That's 8 operations at cost 1, 1 operation at cost 9, and 7 operations at cost 1.

Performing 32 push operations on an array of capacity 8, when doubling array capacity, would be a total cost of $8 + 9 + 7 + 16 + 16 = 56$. That's 8 operations at cost 1, 1 operation at cost 9, 6 operations at cost 1, 1 operation of cost 16, and 16 operations of cost 1.

The big-oh complexity for this push operation as N gets very large is $O(1+)$.

2. Performing 16 push operations on an array of capacity 8, when adding 2 memory blocks, would be a total cost of 60.

Performing 32 push operations on an array of capacity 8, when adding 2 memory blocks, would be a total of 260.

The big-oh complexity for this push operation as N gets very large is $O(n)$.

3. Instead of shrinking the array by one half every time you get below that point, perhaps only shrinking by a quarter would suffice. For instance, imagine you have an array of 100 elements. Say you have popped your way down to only 50 elements. Rather than free the empty 50 elements your not using, just free 25 of them. This way, you avoid having to reallocate memory so quickly if you were to perform a push operation at the 50 mark.