Proof of O(1) amortized Time Complexity for Push and Pop operations

Rajarshee Das 2022CS11124

1. For Push operation:

Let the starting size of array be n.

Cost of pushing 1 operation till $(n)^{th}$ element = 1

Cost of pushing $(n+1)^{th}$ element = (2n) + (n) + 1

Explanation: 2n for making a new array of size 2n, then n for copying the previous n elements is n and then 1 for pushing the $(n+1)^{th}$ element.

Therefore, total cost for n+1 push operations = n*1 + 2*n + n +1 = 4*n +1

Therefore, the time complexity for n push operations is O(4*n+1) = O(n)

So, amortized time complexity, i.e., average time complexity for n+1 push operations is = O(n)/n = O(1).

Hence, proved.

2. For Pop operation:

Let the initial size of the array be n.

Till n/4 each pop operation's cost will be 1.

At $(n/4 -1)^{th}$ pop the total cost will be = (n/2) + (n/4 -1)

Explanation: For making a new array of half the size cost will be n/2; and then for copying the n/4 -1 elements the cost will be n/4 -1
Therefore total cost = 3*n/4 + n/4 -1 + n/2 = 5*n/4 -1

Considering this resize to be last resize of the dynamic array (Since, the minimum capacity of the array needs to be 1024), the total time complexity = O(5*n/4 -1) = O(n)

Therefore, the amortized time complexity, i.e., the average time complexity for each element = O(n)/n= O(1)

Hence, Proved. For the remaining operations it is O(1) for each operation.