Stack – Alternative Approach

Let us improve the complexity by using the array doubling technique. If the array is full, create a new array of twice the size, and copy the items. With this approach, pushing `n` items takes time proportional to n $(not\ n^2)$.

For simplicity, let us assume that initially we started with $n = 1$ *and moved up to* $n = 32$. *That means, we do the doubling at* $1, 2, 4, 8, 16$. *The other way of analyzing the same approach is: at* $n = 1$, *if we want to add (push) an element, double the current size of the array and copy all the elements of the old array to the new array.*

At $n = 1$, *we do 1 copy operation at* $n = 2$, *we do 2 copy operations, and at* $n = 4$, *we do 4 copy operations and so on. By the time we reach* $n = 32$, *the total number of copy operations is* $1 + 2 + 4 + 8 + 16 = 31$ *which is approximately equal to* $2n$ *value*(32).

We can see that the series we get is :

$$\mathbf{2^0 + 2^1 + 2^2 + 2^3 + 2^4 = 31}$$

which is approximately equal to $\mathbf{2}$***n value***$(\mathbf{32})$.
If we observe carefully, we are doing the doubling operation \`$\log n$\` ***times. Now, let us generalize the discussion. For*** \`***n***\` ***push operations we double the array size*** \`***logn***\` ***times. That means, we will have*** \`***logn***\` ***terms in the expression below. The total time*** $T(n)$ ***of a series of n push operations is proportional to***:

$$\mathbf{1 + 2 + 4 + 8 + \cdots + \frac{n}{4} + \frac{n}{2} + n}$$
$$\mathbf{= n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots + 4 + 2 + 1}$$
$$\mathbf{= \frac{n}{2^0} + \frac{n}{2^1} + \frac{n}{2^2} + \frac{n}{2^3} + \cdots + 2^2 + 2^1 + 2^0}$$
$$\mathbf{= \sum_{i=0}^{n} \left(\frac{n}{2^i}\right)}$$
$$\mathbf{= n + \sum_{i=1}^{n} \left(\frac{n}{2^i}\right)}$$

A geometric sequence has a constant ratio r and is

$defined\ by\ a_n = a_1 \times r^{n-1}$

$$\sum_{i=1}^{n} \left(\frac{n}{2^i}\right)$$

$$a_i = \frac{n}{2^i}\ , a_{i+1} = \frac{n}{2^{(i+1)}}$$

$$Computing\ the\ adjacent\ ratio: r = \frac{a_{i+1}}{a_i}$$

$$r = \frac{\frac{n}{2^{(i+1)}}}{\frac{n}{2^i}} = \frac{n}{2^{(i+1)}} \times \frac{2^i}{n} = \frac{1}{2^{(i+1-i)}} = \frac{1}{2}$$

$$when\ i = n\ , then:$$

$$a_i = a_1 r^{i-1}$$

$$a_1 = \frac{n}{2}\ and\ r = \frac{1}{2}\ , hence:$$

$$a_i = \frac{n}{2^i}\left(\frac{1}{2}\right)^{i-1}$$

$$a_i = \frac{n}{2^i} \times \frac{1}{2^{i-1}}$$

$$a_i = \frac{n}{2^i} \times \frac{1}{2^{i-1}}$$

$$a_i = \frac{n}{2^{i-1+1}} = \frac{n}{2^i}$$

$$Geometric\ sequence\ sum\ formula:$$

$$S_n = a_1 \times \frac{1 - r^i}{1 - r}, where\ r \neq 1$$

$$Now\ put\ the\ values:$$

$$i = n, a_1 = \frac{n}{2}, r = \frac{1}{2}$$

$$= \frac{n}{2} \times \frac{1 - \left(\frac{1}{2}\right)^n}{1 - \frac{1}{2}}$$

$$= \frac{n\left(1 - \left(\frac{1}{2}\right)^n\right)}{2\left(1 - \frac{1}{2}\right)}$$

$$= \frac{n\left(1 - \left(\frac{1}{2}\right)^n\right)}{2\left(\frac{1}{2}\right)}$$

$$= n\left(1 - \left(\frac{1}{2}\right)^n\right)$$

$$Hence, \sum_{i=1}^{n} \left(\frac{n}{2^i}\right) = n\left(1 - \left(\frac{1}{2}\right)^n\right)$$

$$And\ , n + \sum_{i=1}^{n} \left(\frac{n}{2^i}\right) = n + n\left(1 - \left(\frac{1}{2}\right)^n\right)$$

$$= n + n - n\left(\frac{1}{2}\right)^n$$

$$= 2n - n\left(\frac{1}{2}\right)^n$$

$$Hence\ , we\ have\ O\left(2n - n\left(\frac{1}{2}\right)^n\right) = O(2n) = O(n)$$

$T(n)$ ***is*** $O(n)$ ***and the amortized time of a push operation is*** $O(1)$.

Here Amortized time is average time taken per operation:

1st push $= 1$ ***element and now size is*** $1 = \frac{1}{1} = O(1)$

Double the array size $= 2$

Copy 1 to new array and insert 1 element say 2 again:

amortized time: $\frac{1+1}{2} = \frac{2}{2} = O(1).$

***Double the array size to* 4.**

***Copy the previous element to new array i.e.* 1, 2**

***Now we can insert further* 2 *more elements into the array*:**

***say* : 3 *and* 4 *are inserted*.**

$$\textit{Hence amortized time} = \frac{4\ \textit{element}}{\textit{size} = 4}\ \textit{i.e.}\frac{4}{4} = O(1).$$

***Hence amortized time of a push operation here is O*(1).**

______________********************_________________