

Bonus: Assignment 2

Amortised time complexity is average time complexity. In Stack_B I have implemented a dynamic array with the capacity parameter $C = 2$. This means:

1. The minimum capacity of the array and its default capacity is 1024.
2. When the array is full, a new array of capacity twice the old one is formed.
3. When the array size is less than or equal to the capacity/4 ; the array capacity is shrunk to one-fourth. This is done at $\frac{1}{4}$ instead of $\frac{1}{2}$ in order to minimize hysteresis.

So pop operation: array copying takes $O(n)$ but then you can pop for other $3n/4$ operations so amortised time is $O(n)/(3n/4) = O(1)$.

For push operation: array copying into $2n$ size array takes $O(n)$ but then you can push for n times. So amortised time is $O(n)/n = O(1)$.

For arithmetic operations: 2 pops, one operation and 1 push so amortised time again has constant steps and hence these operations are also $O(1)$.

Getters: These are just returning the stored class values and hence this access is $O(1)$.