1. Explain the difference between array size and capacity.

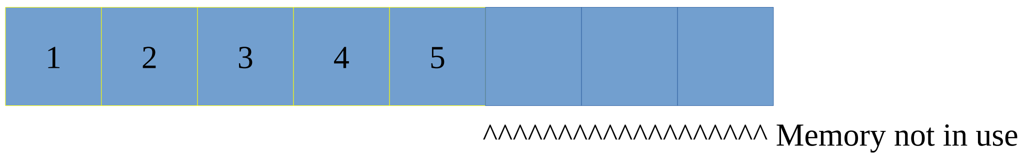
Array size is the current number of elements inside the dynamic array, as placed there by the user. The array capacity is the amount of memory allocated by the array that can eventually be used to store elements.

2. What happens when an array needs to grow beyond its current capacity? Explain and produce a diagram showing the memory layout before and after expansion.

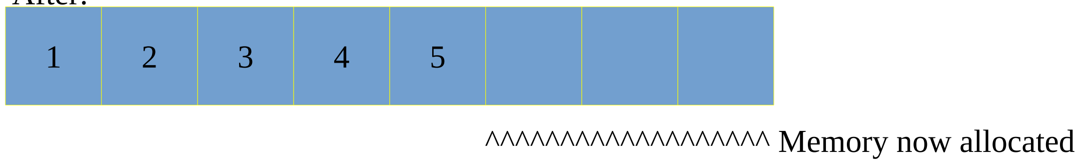
1. Case: There is space in memory after the end of the array

In this case, the array can simply resize itself to include the space after the end of the array since it is not in use. No copying needs to be done as it is starting at the same memory location.

Before:



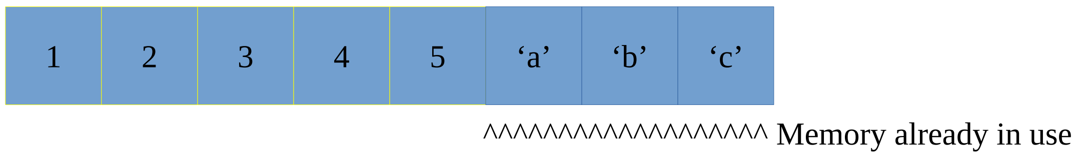
After:



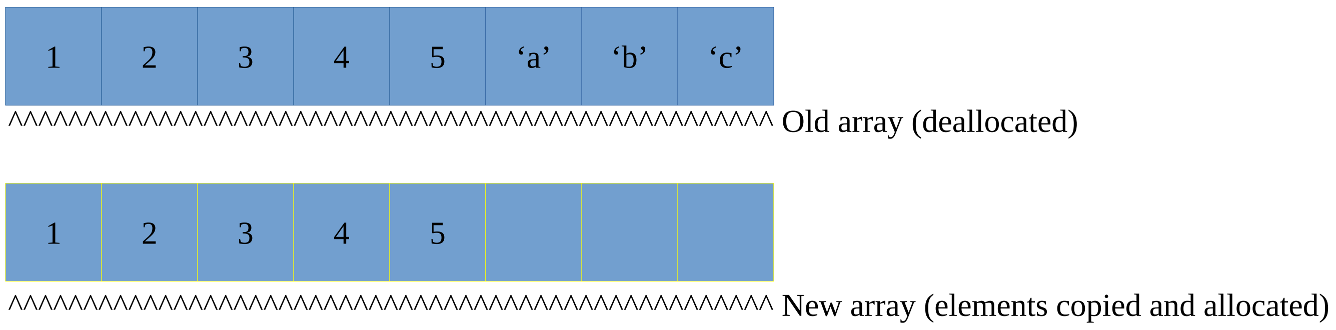
2. Case: Memory after array is already occupied

In the case that the memory after the array is already occupied the allocator will have to find a new position in memory that fits the array at its new capacity and copy over the elements from the original array over to the new one before deallocating the old array memory.

Before:



After:



3. Discuss one or more techniques real-world array implementation use to amortize the cost of array expansion.

One of the most prominent techniques to amortize the cost of array expansion is to use a growth factor. For instance, if the the array does not have enough capacity to add another element, the capacity will grow by a factor larger than 1. This allows the program to expand the array less at the cost of higher memory usage.