1:
An array's size describes the current number of data elements it contains, and its capacity is the maximum number of elements it can store. For example, an array could have 6 current elements giving it a size of six but 20 elements of space reserved in memory

2: When an array needs to expand its capacity there are one of two cases that occur. The first is when there is space in memory directly after where the array is stored in which case said memory can be reserved for the array. The other is when the memory directly after the array is already occupied, in which case the entire array will be moved to a new point in memory where there is enough space to add the required capacity to the end of the array.

3: One way that real-world arrays amortize the cost of array expansion is by increasing the size by more than required whenever it is necessary. Typically, when trying to append to a full array instead of making and array that is one element longer and then copy over the required data, we create an array that is double the size. Because appending to a none full array takes O(1) time and appending to a full array takes O(n) time, this dramatically reduces the time required for future appends until the array is full again