1. Explain the difference between an array size and capacity [0.1 pts]

Array size references the number of elements actively in the array, whereas capacity is the number of elements it can actually hold.

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. First, consider the case where there is space in memory after the end of the array [0.1 pts]

In the case where there is enough space in memory all that would happen is that the area of memory allocated for the array would be increased to handle the new data coming in.

2. Then, consider the case where the memory after the end of the array is occupied by another variable. What happens in that case? [0.2 pts]

The memory of the original array would be copied somewhere else in memory that is capable of handling its memory requirement. However, this can be problematic as it can incur a time complexity of O(n).

3. Discuss one or more techniques real-world array implementations use to amortize the cost of array expansion [0.1 pts]

Real-world techniques used in array implementations are many in number, however the most used one I found was the “Doubling Strategy” for geometric growth. What this technique does is that whenever it expands an array instead of just reallocating the memory of the original array and increasing it for the new data, it simply just doubles the array capacity. While this is costly short-term, this strategy results in infrequent reallocation for the array making it better long-term.