

In class we discussed the inner functioning of dynamic array

1. Explain the difference between an array size and capacity [0.2 pts]
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 [0.3 pts]
  2. Then, consider the case where the memory after the end of the array is occupied by another variable. What happens in that case? [0.3 pts]
  3. Discuss one or more techniques real-world array implementations use to amortize the cost of array expansion [0.2 pts]

1. The difference between an array size and capacity is that the size of the array means how many elements are in the array. Whereas capacity means how many elements the array can hold in total, not just what is in it already.
2. If there is space in memory after the end of the array, then the array will just increase its capacity by taking the next spot of memory after the current one

example:

Before expansion

1	3	2	4	5	Unused free memory	Unused free memory
0x00	0x04	0x08	0x0c	0x10	0x14	0x18

After expansion:

1	3	2	4	5	7	8
0x00	0x04	0x08	0x0c	0x10	0x14	0x18

## 2.2

In the case that the memory at the end of the array is occupied, then it would copy the whole array into another section of memory where there is enough space.

Example:

Before expansion

1	3	2	4	5	Memory taken up by var.	Memory taken up by var.
0x00	0x04	0x08	0x0c	0x10	0x14	0x18

After expansion (the address in memory has changed to a location where enough space is available)

1	3	2	4	5	7	8
0x40	0x44	0x48	0x4c	0x50	0x54	0x58

3. One way is that it expands the array on initialization by a factor of greater than 1 (1.125, 1.5 or 2), so that there is extra space at the end of the array so that it most likely will not need to copy the whole array into another segment of memory.