Lab 2 – Dynamic List Resizing

Name: **Asrar Syed**

Course Section: **CSC 2720-012**

Panther ID: **002679083 🡪 N = 70**

1. Provide a line-by-line analysis of the time and space complexity of your *arr\_resize()* method.

**Time Complexity**

* Creating and initializing a new list (line 7 and line 9), both have a time complexity of O(n).
* Slicing and copying the list (line 10), has a time complexity of O(n).
* Printing out the array (line 11), also has a time complexity of O(n)

**Space Complexity**

* The space complexity for the lists on (line 7 and line 9) is O(n).
* I used a temporary slice I use to copy elements (arr[:len(new\_arr)] on line 10) takes up similar space to the length of the slice, so a time complexity of O(n).

Overall a Time Complexity and Space Complexity of O(n).

2. If N elements are pushed to the end of a dynamic array that has initial size 0, capacity 8, and growth factor = 1.25, how many times is the dynamic array resized?

Since N = 70, it will take 10 resizes to go from a capacity of 8 to 87, enough to hold 70 elements. (approx. through just rounding it)

If not rounding then 11 resizes.

3. If N elements are pushed to the end of a dynamic array that has initial size 0, capacity 8, and growth factor = 1.25, how many cells of the final array are empty?

Since N = 70, then 87-70 is 17. So this means 17 empty cells

If not rounding then 23 empty cells.

Here, N = SUM\_OF\_THE\_DIGITS\_OF\_YOUR\_PANTHER\_ID\_NUMBER \* 2

For example, if your panther id # is 002-22-3311, then consider N = 28.