

CSCI 102 assignment 1 – Dynamic Arrays

Feb 3

Recall that in our implementation of `ArrayList`, we were limited in how many elements we could add to the list. In this problem, we will overcome this limitation by implementing a dynamic array – this is the implementation that Python for example uses with its `list` objects. The idea will be that if we want to add an element but `array` is full then we will 1) create a new array `bigger_array` with double the size of `array`, 2) copy all the elements from `array` into `bigger_array`, 3) set this new array to be `array = bigger_array`, and 4) add an element to this array. Please submit a java project with code that does the following. You may copy any code on the class website.

- Create a new class `DynamicList` that is a subclass of `ArrayList`. Give it new attribute `int capacity` that keeps track of the size of the array.
- Create a constructor for `DynamicList` that takes as input the starting size of `array`. You should modify the code of `ArrayList` to do so. Give `DynamicList` a default constructor that sets the size of `array` to 1000.
- Create a new method in `DynamicList`, `private void increaseArraySize(int new_size)` that creates a new array of size `new_size`, copies the contents of `array` into this new array, reassigns `array` to this new larger array, and updates the value of `capacity`.
- Override the methods `addFirst` and `addLast` so that if `size==capacity` then you double the size of `array` before adding a new element. Use the `super` keyword to avoid rewriting the entire method.
- Include a main method that creates a `DynamicList` with an array of capacity 2, adds the numbers 1 to 20 to the end of `DynamicList` and then prints `capacity`.

Answer the following questions and submit them in a separate file:

- What is the benefit of `DynamicList` over `DoublyLinkedList`?
- What is the asymptotic complexity of `increaseArraySize` with respect to the parameter `new_size`?
- If instead of doubling the size of the list when `size==capacity`, we instead increased the size by 1 by calling `increaseArraySize(size+1)`, what is the complexity of the method `addLast` when the array is full? In this case, what is the complexity of adding N elements to the end of the list?
- (Bonus) If N is a large power of 2, $N = 2^M$ then what is the complexity of adding N elements to the end of an array in the code we have written? Can we make the asymptotic complexity of our code better by incrementing the size of `array` in larger steps, say by multiplying `capacity` by 4 instead of 2?

Please submit your code and answers to the questions in a zipped folder on brightspace by Feb 10. Remember to use the principles of encapsulation and least privilege!