CS II Data Structures

Resizing ArrayList experiment

In this activity, you'll explore the running time of inserting elements into an ArrayList. You must work in groups of 2-3.

Roles:
- **Driver**: follows the instructions on their computer
- **Scribe**: writes down answers the questions
- **(optional 3rd person) Skeptic**: double-check your group's answers as you go

Part 1: ArrayList that increases its capacity by 1
1. The following code *simulates* insertion into an ArrayList; when the underlying array is full, we copy the elements to an array of +1 size. In previous experiments (such as in Lab) you ran code and timed how long it took to finish, so in this case, what do we mean by *simulate*?

```python
steps = 0
# initially ArrayList's internal array is size 4
arraysize = 4
for n in range(1, 1001):
    if n > arraysize:
        # ArrayList is full so n steps to copy to new array of size n+1
        steps += n
        # new array has +1 size
        arraysize += 1

    # 1 step to copy the new value into the last spot
    steps += 1

print str(n)+" \t"+str(steps)
```

2. What will each print statement print out?

Go to [https://repl.it/LkYB/1](https://repl.it/LkYB/1) to run the code (you do not need an account)
3. Let's see the results. Copy the printed data to your clipboard.


You should see the data.
Choose scatterplot, X: A, Y: B, then view your result in the plot window!

4. What are the X and Y axes? (your answer to #2 may help remind you). Also, label the axes in the web app and save a screenshot of your plot.

5. What does the plot tell us about inserting into the ArrayList?

Part 2: ArrayList that increases its capacity by 2 times

1. The following code is like the previous but with a small change. It simulates inserting elements into an ArrayList; when the underlying array is full, we copy all the elements to an array of twice the size.

steps = 0
# initially ArrayList’s internal array is size 4
arraysize = 4
for n in range(1, 1001):
    if n > arraysize:
Resizing ArrayList experiment

```python
# ArrayList is full so n steps to copy to new array of size 2n
steps += n
# new array has x2 size
arraysize *= 2

# 1 step to copy the new value into the next open spot
steps += 1

print str(n)+"\t"+str(steps)
```

2. Go to https://repl.it/LkY8/1 to see and run the code. (you do not need an account). Copy/paste the results into a new browser tab of https://plot.ly/create/ (you do not need an account). Label the axes of the plot and save a screenshot.

3. There are jumps in the plot. What do these jumps correspond to?

4. What does the plot tell us about inserting into the ArrayList that doubles its size when it is full?

5. Compare the plots in Part 1 and Part 2. What conclusions can you draw about the two ways to handle a full array in the ArrayList implementation?


7. (bonus) What is the worst case running time of performing N insertions into a Part1 ArrayList? A Part2 ArrayList?