Implement BubbleSort and Quicksort for integers

- Count the number of assignments
- Count the number of conditionals

```
def bubblesort(some_list):
    if len(some_list) == 0:
        return([])
    iteration = 1
    while iteration < len(some_list):
        for i in range(len(some_list)-iteration):
            if some_list[i] > some_list[i + 1]:
                temp = some_list[i]
                some_list[i] = some_list[i + 1]
                some_list[i + 1] = temp
        iteration += 1
    return(some_list)
```

2 loops for main part of bubblesort:
Outer ‘while’ loop executes n times:

- 1 assignment

Inner ‘for’ loop content executes between n and 1 times depending on the iteration and previously ordered numbers:

- 1 conditional
- 3 assignments

```
def quicksort(some_list):

    if len(some_list) == 0:
        return([])

    lesser = []
    equal = []
    greater = []

    if len(some_list) > 1:
        pivot = some_list[int(len(some_list)/2)]
        for i in some_list:
            if i < pivot:
                lesser.append(i)
            if i == pivot:
                equal.append(i)
            if i > pivot:
                greater.append(i)
        return(quicksort(lesser)+equal+quicksort(greater))
    else:
        return(some_list)
```

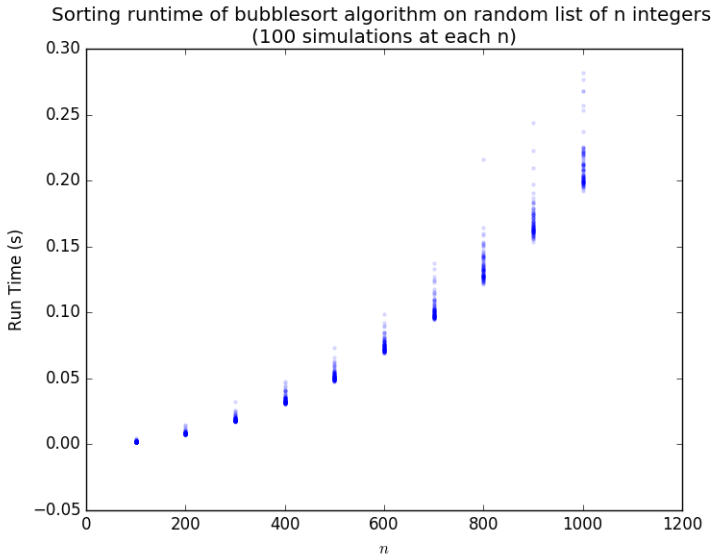
1 for loop within a recursive loop for main part of quicksort:
Recursive loop:

- 1 conditional
- 1 assignment

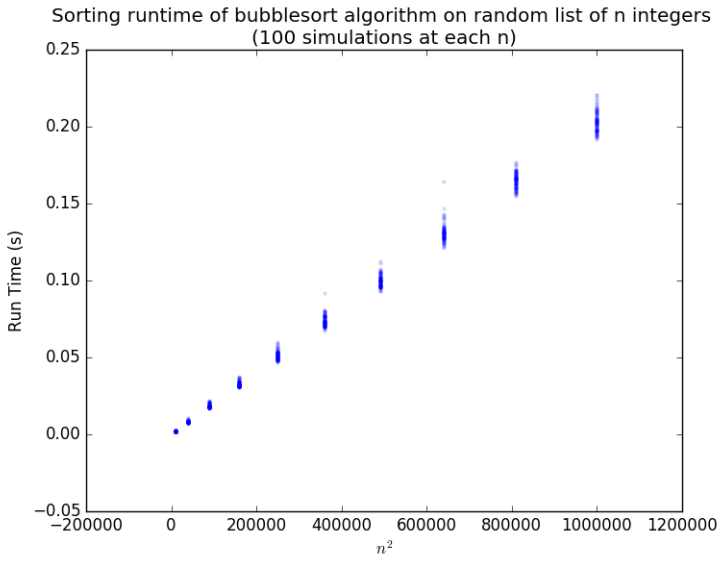
For loop:

- 1 assignment for each element in the current list/sublist
- Best case scenario, this for loop is accessed about $\log_2 n$ times (perfect binary division)
- Worst case scenario, this for loop is accessed n times

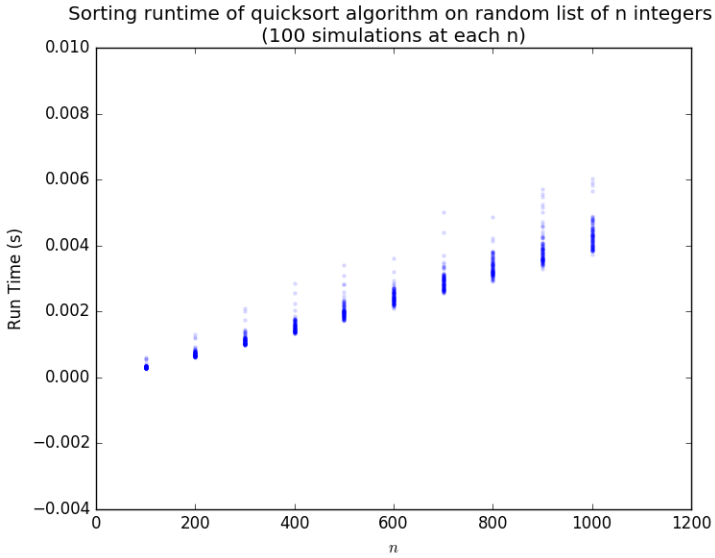
BubbleSort is O(n^2) on average – outer for loop is of order n, inner for loop is order n



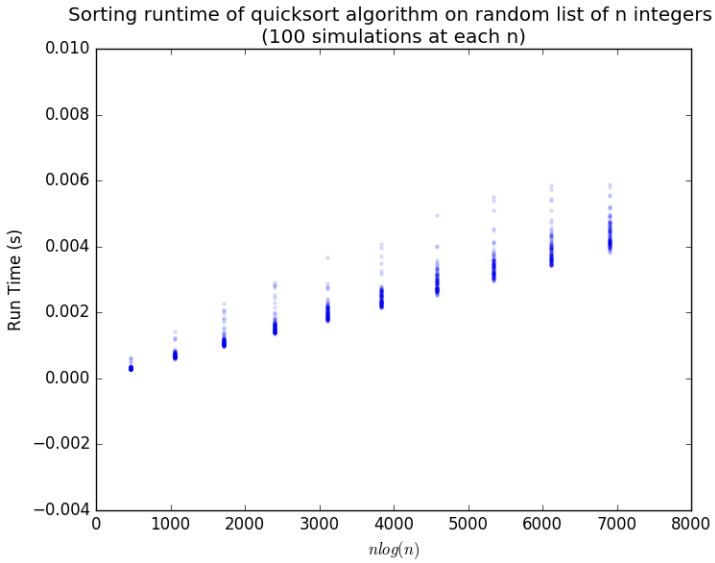
Transforming the x-axis to from n to n^2) shows that runtime is a linear function of n^2:



QuickSort is O(n log(n)) on average – recursively splitting the list is of order log(n) and reordering each sub-list is of order n



Transforming the x-axis to from n to nlog(n) shows that runtime is a linear function of nlog(n):



Code available on GitHub:

https://github.com/TownJasonP/BMI203_HW1