CS 1301 - Introduction to Computing
Fall 2017
Homework 9: Big O – Searching – Sorting

Rules:

- You must upload your submissions through gradescope.
  - Login into gradescope.
  - Select CS1301
  - Select Homework 09
  - You must select the "**SUBMIT PDF**" option.
  - Submit **HW09.pdf**
- This is an individual assignment. No collaboration is permitted.
- Good handwriting is encouraged and if a TA cannot read a problem it could result in a zero
- Due Date: **Thursday, November 9th 11:55PM.**

Name: _____ GTLogin: _____ Section_____

1. [20pts]: For each of the following pieces of code, write down the time complexity that the code will run in, choosing from `O(1), O(log n), O(n), O(n log n), O(n^2), O(n^3)`:

```
for i in range(n):
   i *= 2
   for j in range(n):
      print(i * j)
```

Big-O:_____

```
for i in range(10):
   for j in range(n):
      print(i-j)
```

Big-O: _____

```
for i in range(n):
   for j in range(n, n/3, -9):
      for k in range(n):
        return n
```

Big-O:_____

```
for i in range(521313*2213*11):
   for j in range(i ** i ** i):
      for y in range(j * i):
         print(i, j, y)
```

Big-O:_____

```
i = 0;
while i < n:
   print("David" * i)
   i *= 2
   i += 1
```

Big-O:_____

```
for i in range(len("McDonald")):
    print("hi")
    j = 0;
    while j < n:
        sum = i + j
        j += 1
        for k in range(sum):
            print(k)
```

Big-O:_____

2. [10pts] Designer Karoline K. (the "other" Kardashian) is having an exclusive fashion show where she has N models. If N=5, the models will be numbered [0,1,2,3,4]. Because all the models are foreign, none of them know each other, so you write the following code to "introduce" each model to every other model.

```
def introduce(ModelA, ModelB):
    print(ModelA, "I'd like to introduce you to", ModelB)
    print(ModelB, "meet", ModelA)

def fashionShow( listOfModels):
    for modelX in listOfGuests:
        for modelY in listOfGuests:
            introduce(guestX, guestY)

fashionShow( [0,1,2,3,4] )
```

Notice that the above code introduces the same models to themselves, and also introduces a pair of model twice (it introduces 0 to 1, and then 1 to 0). This is not exactly the same as a fashion show with real humans.

Your question: If you assume that a call to the introduce(...) function is your unit of work (i.e. just like a comparison in a sorting algorithm), what is the Big O complexity class of this problem? In other words, as the number of models (N) increases, how quickly does the number of introductions increase?

Answer this question by filling in the following blanks:

```
If N = 2  the number of Introductions =     ___________

If N = 4  the number of Introductions =     ___________

If N = 8  the number of Introductions =     ___________
```

So therefore, the complexity class is: O(_____)

Also, if it takes 1 second to introduce each pair of models, how many seconds will you spend doing introductions if you have 50 models?

Answer: _____

3. [5pts] Given the following list, list the elements in the order in which binary search accesses them when searching for the number 4 (if an element is not accessed/compared then don't list it). Note: if necessary, the middle of an even sized list will be the lower index number e.g. the

middle of [1, 2, 3, 4] would be 2.

```
[1, 4, 9, 15, 32, 99, 107]
```

4. [5pts] Would you use binary search or linear search to search through the following list for some number? Why?

```
[17, 3, 81, 62, 19]
```

5. [18pts] Identify the algorithm being used to sort each of the following lists, and finish sorting the list showing the new list at each step of the algorithm. Show a complete iteration, not the small substeps.

Algorithm A

Original List: `[7, 3, 4, 2, 1]`
First iteration: `[3, 4, 2, 1, 7]`
Second iteration: `[3, 2, 1, 4, 7]`

Third iteration: _____

Fourth iteration (if needed, otherwise leave blank): _____

Name of Algorithm A: _____

Big O of Algorithm A: _____

Algorithm B

Original List: `[4, 1, 6, 7, 2]`
First iteration: `[1, 4, 6, 7, 2]`
Second iteration: `[1, 2, 6, 7, 4]`

Third iteration: _____

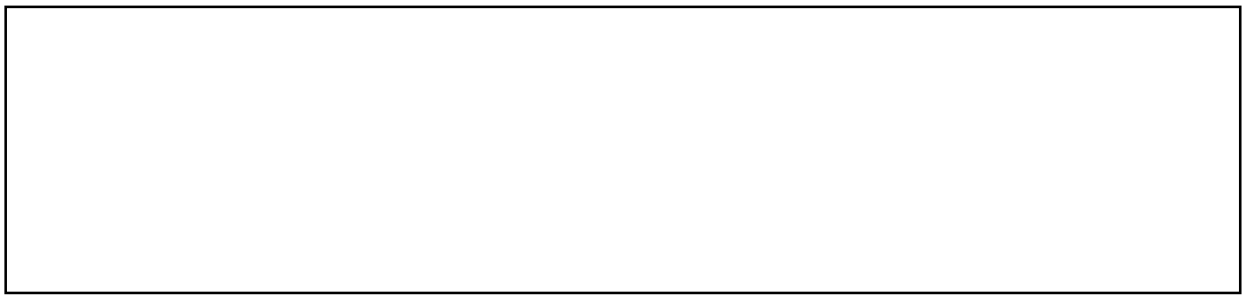
Fourth iteration (if needed, otherwise leave blank): _____

Name of Algorithm B: _____

Big O of Algorithm B: _____

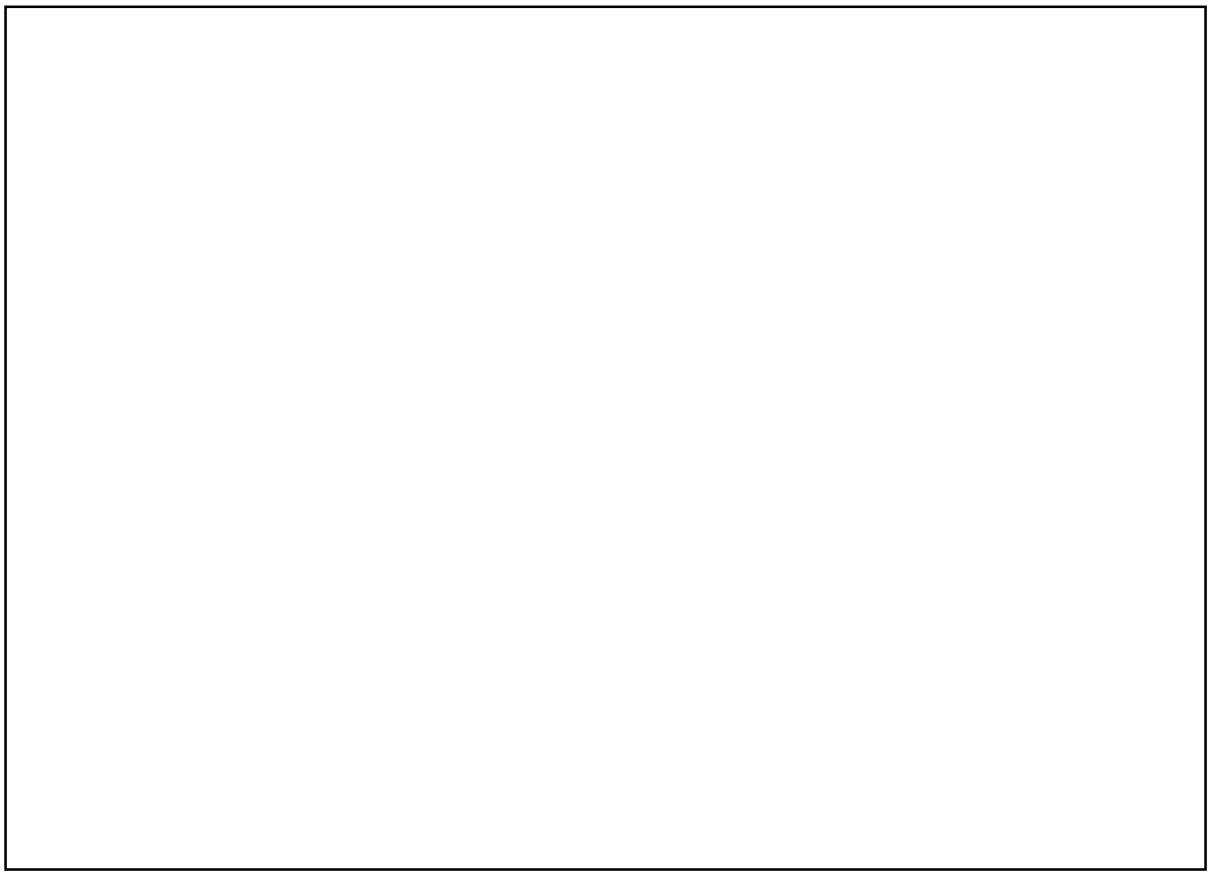
6. [7pts] Given a properly implemented merge sort algorithm and the list `[8, 4, 2, 1, 7,11]` is it possible for the merge sort algorithm to eventually have to merge the following two lists? **Why or why not?**

`[8, 4, 2] [1, 7, 11]`



7. [20pts] Draw a diagram that illustrates how the merge sort algorithm would sort this list. Draw the contents of the list after each splitting and merging step of the algorithm. (Note: you can split the halves either way, but make sure you stay consistent):

```
[5, -9, 1, 4, 0, -4, 3]
```



8. [15pts] Here is a sequence of numbers: `3, 8, 2, 6, 0, 11, 1`

(a) [5 pts] Illustrate how a bubble-sort would sort the above list of numbers. After each pass, underline the positions that are guaranteed to be in sorted order. Do all passes, do NOT make short-cutting optimizations.

(b) [5pts] Illustrate how a selection-sort would sort the above list of numbers. After each pass, underline the numbers that are guaranteed to be in sorted order.

(c) [5 pts] Illustrate how a merge-sort would sort the above list of numbers.