Assignment: Asymptotic Notations and Correctness of Algorithms

[You may include handwritten submission for the parts of the assignment that are difficult to type, like equations, rough graphs etc., but make sure it is legible for the graders. Regrade requests due to the illegible parts of the work will not be accepted.]

1. **Identify and compare the order of growth**: Identify if the following statements are true or false. Prove your assertion using any of the methods shown in the exploration. Draw a rough graph marking the location of c and n_0, if the statement is True. [A generic graph would do for this purpose. You **don't** have to find the values of c and n_0. On the graph you can just write c and n_0 without mentioning their values. The idea is that you know how it looks graphically.].

   a. $n(n+1)/2 \in O(n^3)$
   b. $n(n+1)/2 \in \Theta(n^2)$
   c. $10n-6 \in \Omega(78n + 2020)$
   d. $n! \in \Omega (0.00001n)$

2. **Read and Analyze Pseudocode:** Consider the following algorithm (In the algorithm, A[0..n-1] refers to an array of n elements i.e. A[0], A[1]... A[n-1])

```
Classified(A[0..n-1]):
    minval = A[0]
    maxval = A[0]
    for i = 1 to n-1:
        if A[i] < minval:
            minval = A[i]
        if A[i] > maxval
            maxval = A[i]
    return maxval - minval
```

   a. What does this algorithm compute?
   b. What is its basic operation (i.e. the line of code or operation that is executed maximum number of times)?
   c. How many times is the basic operation executed?
   d. What is the time complexity of this algorithm?

3. **Using mathematical induction prove below non-recursive algorithm:**

```
def reverse_array(Arr):
    n = len(Arr)
    i = (n-1)//2
    j = n//2
    while(i>= 0 and j <= (n-1)):
        temp = Arr[i]
        Arr[i] = Arr[j]
        Arr[j] = temp
        i = i-1
        j = j+1
```

a. Write the loop invariant of the reverse_array function.
b. Prove correctness of reverse_array function using induction.

----------------------(Ungraded question: you can try this question if time permits)---------------

Any number greater than 8 can written in terms of three or five.

a. Write a pseudocode of algorithm that that takes a number greater than 8 and returns a tuple (x,y); where x represents number of threes and y represent number of fives make that number
If number = 8 your pseudocode should return (1,1)
b. Code your pseudocode into python and name your file ThreeAndFive.py