1. **(2 pts)** Determine asymptotic growth rate (Big-O) of the following functions and order them in the ascending order (fastest to slowest).
   $7n\log n + 5n$
   $2^{\log n}$
   $5n+50\log n$
   2^n
   $3n$
   2^{12}
   $n^2 + 100n$
   $15n\log n$
   n^3

2. **(2 pts)** Characterize following Python function using Big-O notation.
   To get full points, justify your answer.

```
def function(data):
    value = data[0]
    for i in range(len(data)):
        if data[i] < value:
            value = data[i]
    return value
```

3. **(2 pts)** Characterize following Python function using Big-O notation.
   To get full points, justify your answer.

```
def sum_of_prefixes(X, Y): # X and Y have the same size
    '''Return number of elements in Y equal to the sum of prefix sums in X'''
    n = len(X)
    counter = 0
    for i in range(n): # loops 0 to n-1
        tot = 0
        for j in range(n): # loops 0 to n-1
            for k in range(1 + j): # loops 0 to j
                tot += X[k]
                if Y[i] == tot:
                    counter += 1
    return counter
```

4. **(4 pts)** Determine computational complexity of the following Python code and prove it using induction.
   To get full points, justify your answer and describe induction steps.

```
n = 10
for x in range(n):
  print("Outer iteration: {}".format(x))
  for y in range(x + 1, 0, -1):
    print("\tInner iteration: {}".format(y))
  print()
```