CSCI 305 Assignment 1

Isaac Boaz

October 11, 2023

Problem 1

```
for i = 1 to n
    j = 1
    while j<= i do
        j = 2.5 * j
```

cost	time
C_1	$n+1$
C_2	n
C_3	$(n+1)\log_{2.5}$
C_4	$n\log_{2.5} n$

Detailed Runtime:

$$C_1 \cdot (n+1) + C_2 \cdot n + C_3 \cdot \lceil (n+1)\log_{2.5} \rceil + C_4 \cdot \lceil n \log_{2.5} n \rceil$$

Asymptotic Runtime:

$$\Theta(n \log n)$$

Problem 2

```
def factorial(n):
    x = 1
    while n > 1:
        x = x * n
        n = n - 1
    return x
```

cost	time
C_1	1
C_2	n
C_3	$n-1$
C_4	$n-1$
C_5	n

Detailed Runtime:

$$C_1 \cdot 1 + C_2 \cdot n + C_3 \cdot (n-1) + \cdot C_4 \cdot (n-1) + C_5 \cdot n$$

Asymptotic Runtime:

$$O(n)$$

Problem 3

```
def Binary_InsertionSort(A):
    for j = 2 to n:
        key = A[j]
        // insert A[j] into the sorted sequence A[1..j-1]
        left = 1
        right = j - 1
        while right > left:
            mid = floor((left + right) / 2)
            if key > A[mid]:
                left = mid + 1
            else right = mid
        if key > A[left]:
            left = left + 1
        for i = j downto (left + 1):
            A[i] = A[i - 1]
        A[left] = key
```

1. Note that regardless of input, the for loop will always run n times, with the contents of the loop running $n-1$ times.
   Worst Case occurs when the array is sorted in reverse order, as the left and right bounds will need to be updated $\log_2 j$ times.
   Best Case occurs when the array is already sorted, as the left and right bounds will only need to be updated once.

2. **Worst Case** for the *while* block is the same in the best and worst case scenario, running $\log_2 j$ times.