# Time Complexity Analysis

1. (1Pts) For each of the following expressions, what is the order of growth (in Θ(Theta) notation) that best describes it?

For example: 10*n → Θ(n)

1) 3n**2 + 7n**3 + 4 Θ(n^3)
2) 10log(n) + 5n Θ(n)
3) 3**n + n**2 Θ(3^n)
4) 100n + nlog(n) Θ(n log(n))
5) 5 + 40 Θ(1)
6) log(n) + 4n Θ(n)

2. (2Pts) Please specify the time complexity for the following programs.

```
def program1():
    list = [1,2,3,4,5,6,7,8]
    even_list = []
    for i in range(len(list)):
        if i % 2 == 0:
            even_list.append(i)

    return even_list
```

Θ(n)

```
def program2():
    list1 = [1,2,3,4,5,7]
    list2 = [3,4,5,7,8,9]

    output_list1 = [i for i in list1]

    output_list2 = []
    for i in list1:
        for j in list2:
            output_list2 += [i,j]
```

```
    return (output_list1, output_list2)
```

$\Theta(n^2)$

```
def program3(n):
    epsilon = 0.01
    low = 0
    high = n
    ans = (high + low) / 2

    while abs(ans**4 - n) >= epsilon:
        if ans**4 > n:
            high = ans
        else:
            low = ans
        ans = (high + low) / 2

    return ans
```

$\Theta(\log n)$

```
def program4(n):
    """ n > 0 """
    answer = 0
    while n > 0:
        answer += n%10
        n = int(n/10)

    return answer
```

$\Theta(\log n)$