

CS240 Homework 2

DUE Friday, Sept 19 at 14:30 (2:30pm)

Your answers to the following exercises should be submitted through Canvas as a single `hw02.pdf` file. Equations should be properly formatted using equation editing software, such as LaTeX or the equation editor in MS Word or LibreOffice.

1. Analyze each of the following three Python functions. In each case you may assume that `values` is a list of integers. The length of `values` should be used as the input size (n). For each function provide both the exact growth rate and the appropriate big-O class. (3pts)

NOTE: Integer addition is the *only* basic operation that you should count when determining the exact operation counts and growth rates for this question.

```
def someFunc1(values):
    sum = 0
    for i in range(len(values)):
        sum += i
    for val in values:
        sum += 4 * val
        sum += 8 * val
    return sum
```

```
def someFunc2(values):
    sum = 0
    for i in values:
        for j in values:
            sum += 2 * j
            for k in range(5):
                sum += k * k
    return sum
```

```
def someFunc3(values):
    sum = 0
    for i in values:
        j = len(values)
        while j > 1:
            sum += 1
            sum += i
            j = j/2
    return sum
```

2. For the following pairs of functions, indicate whether the ?? could be replaced with O , Ω or Θ . More than one may be correct: indicate all that apply. (2pts)

	$f(n)$	$g(n)$	$f(n) \in ??(g(n))$
a)	$2n$	n^2	
b)	n	$\log_2 n$	
c)	$.001n^3$	$100n^2 + 1000n$	
d)	n^2	2^n	
e)	$5n$	$8n + 1$	
f)	$n!$	$n^4 + 10n^2 \log_2 n$	
g)	n	$\log_2(2^n)$	

3. List each of the functions from the table above from slowest to fastest growing. Indicate which functions are in the same complexity class (same big- Θ). (2pts)

Complexity Class	Functions from slowest to fastest growing

NOTE: You may need fewer rows than provided.

4. Consider the following two algorithms: Algorithm A requires $3n + 4$ steps to complete on an input of size n . Algorithm B requires n^2 steps. For what values of n should we prefer algorithm A? For what values of n should we prefer algorithm B? Justify your answer. (1pt)
5. Using either definition of big- Θ , demonstrate that $n^3 + 2n^2 + 3n \in \Theta(n^3)$ (2pts)