

## CSC 6013 – Summer 2022

### Worksheet for Week 4

1) Complete this table rounding each decimal to the nearest integer. This should give you a sense of the comparative growth rates of the functions. Note that  $lg(n)$  means python's  $\log_2(n)$ .

$lg(n)$	$\sqrt{n}$	$n$	$nlg(n)$	$n^2$	$n^3$	$2^n$	$n!$
		1					
		2					
		3					
		4					
		5					
		6					
		7					
		8					
		9					
		10					

For #2, 3, 4, and 5, solve each summation in two ways: write out all terms of the sum and perform the arithmetic, and use one of the formulas in the class notes to confirm your answer.

2)

$$\sum_{i=1}^{15} i^2$$

3)

$$\sum_{i=0}^{10} 2^i$$

4) Round your answer to three decimal places.

$$\sum_{i=1}^8 2^{-i}$$

5)

$$\sum_{i=1}^{n-1} i$$

Asymptotic analysis for #6 and #7: Determine the Big-Oh class of each algorithm. That is, **formally** compute the worst-case running time as we did on class using a table to produce a function that tracks the work required by all lines of code. Include all steps of the algebraic simplification, but you do not need to provide comments to justify each step.

6) Arithmetic mean = “add them all up, and divide by how many”. Let line 7 be the crucial unit of work that you count. Let the size of the problem =  $n$  = the number of entries in the array.

```
1      # Input: an array A of real numbers
2      # Output: the arithmetic mean of the entries in the array
3      def arithmeticMean(A):
4          sum = 0
5          count = 0
6          for x in A:
7              sum = sum + x
8              count += 1
9          average = sum/count
10         return average
```

7) Sum of entries in an upper triangular  $n \times n$  array. Let the size of the problem =  $n$  = the dimension of the  $n \times n$  matrix.

```
1      # Input – an upper triangular square matrices A where all entries below the diagonal = 0,
2      #      and an integer n giving the dimension of this  $n \times n$  matrix
3      # Output – a real number giving the sum of the entries
4      def UpperTriangularMatrixSum (A, n):
5          sum = 0
6          for i in range(0, n):
7              for j in range(i, n):
8                  sum = sum + A[i][j]
9          return sum
```