# **Tutorial for Section 1.2**

# Fundamentals of the Analysis of Algorithm Efficiency

**Exercise 1**

For each of the following algorithms, indicate: (i) a natural size metric for its inputs. (ii) its basic operation (iii) whether the basic operation count can be different for inputs of the same size:

1. Computing the average of *n* numbers
   1. n – this represents the number of elements needed to find the average
   2. To find the average we would sum all values of n then divide by the number of n
   3. No, because you will always need to sum n values and divide by the number of values
2. Computing
   1. n – this represents the element of input to calculate n/n!
   2. Operations used are division and multiplication, which divides n elements by the factorial of n
   3. No, because the count will be the same for elements n of the same size
3. Finding the smallest element in a list of *n* numbers
   1. n – the number of elements in the list
   2. The main operation is a comparison between n elements to find the smallest (<)
   3. Depending on the order of elements in the list, the comparison between elements may differ in the number of times the operation is run (the count). So yes, the count can be different for the same number of elements
4. Reverse display a list of *n* numbers
   1. n – the number of elements in a list
5. Reverse a list of *n* numbers
   1. n – the number of elements in the list

**Exercise 2**

Consider the definition-based algorithm for finding the difference between two matrices. What is its basic operation? How many times has it performed as a function of the matrix order *n*? As a function of the total number of elements in the input matrices?

**The basic operation is subtraction, and it is performed as a matrix n × n times. As a function of the total number of elements with two matrix it would be the sum of n × n for matrix 1 and sum of n × n for matrix 2 and then calculate the difference between the two products.**

**Exercise 3**

Prove that the number of bits in the binary representation of a positive integer *n* is .

**Hint 1**: Consider the smallest and largest integers *n* having *b* bits.

**Hint 2**:

**Exercise 4**

Gaussian elimination, the classic algorithm for solving systems of *n* linear equations in *n* unknowns, requires about multiplications, which is the algorithm’s basic operation.

1. How much longer should you expect Gaussian elimination to work on a system of 1000 equations versus a system of 500 equations?
2. You are considering buying a computer that is 1000 times faster than the one you currently have. By what factor will the faster computer increase the sizes of systems solvable in the same amount of time as on the old computer?

**Exercise 5**

For each of the following functions, indicate how much the function’s value will change if its argument is increased threefold.

1. *n*
2. *n*3
3. *n*2
4. *n*!
5. 2*n*

**Hint**: Use either the difference between or the ratio of and .

**Exercise 6**

***Invention of chess***

1. According to a well-known legend, the game of chess was invented many centuries ago in northwestern India by a certain sage. When he took his invention to his king, the king liked the game so much that he offered the inventor any reward he wanted. The inventor asked for some grain to be obtained as follows: just a single grain of wheat was to be placed on the first square of the chess board, two on the second, four on the third, eight on the fourth, and so on, until all 64 squares had been filled. If it took just 1 second to count each grain, how long would it take to count all the grain due to him?

**Hint**: Use the formula

1. How long would it take if instead of doubling the number of grains for each square of the chessboard, the inventor asked for adding two grains?