

# COMP1201 - Tutorial 3

Daniel Best (Student ID: 29777127)

May 16, 2019

## Question 1

Since the hash function is calculated by means of a modulo operation, the hash value is going to be between the values of 0 and the value of divisor minus 1. This means that the specifics of the actual hash function itself ( $2d_1 + 3d_2 + 5d_3 + 7d_4 + 11d_5$ ) are irrelevant to this problem.

In this case, the divisor is 47, meaning the returned value of the function can be anywhere between 0 to 46:

$$n \% 47 = \{0, 1, 2, \dots, 45, 46\}$$

By means of the Pigeonhole Principle, we can therefore deduce that any non-negative integer  $n$  where  $n < 100000$  will equate to a number from 0 to 46 - i.e. there are 47 possible *pigeonholes*.

Therefore, if there we were to select 2000 numbers within the range specified, these numbers would have to fit into one of these *pigeonholes* - these are the *pigeons*. By dividing 2000 by 47, we can determine how many *pigeons* can be split evenly into each *pigeonhole*:

$$2000 \div 47 \approx 42.55$$

As  $42.55 > 42$ , this means that at least one such positive integer  $x$  must be equal to a hash value that has 43 numbers or more associated with it, with these numbers being in the set of the selected 2000. In other words, there must exist a *pigeonhole* (hash value) that contains 43 or more *pigeons* (the 2000 numbers).

For this statement to be false, the maximum amount of numbers that can be selected are  $47 \times 42 = 1974$  numbers, and in this case there would be exactly 42 *pigeons* in each *pigeonhole*.

## Question 2

For this question, the first step is to determine how to translate the 3 characters that are present in the diary into something that can be interpreted by the compression algorithm.

As the algorithm takes a binary sequence as an input, the simplest such way of doing this is to convert each character into binary. This can simply be achieved by converting each character into two binary bits - as an example:

$$Circle = 00 \mid Square = 01 \mid Triangle = 10$$

Now that each of these symbols has been converted to an input that the compression algorithm can interpret, the required running time complexity can be calculated from this. By default, the algorithm has a time complexity of  $3n^5 + 5n^3 + 1$ , where  $n$  is a binary sequence of length  $n > 0$ .

To determine the new time complexity of the algorithm, the value of  $n$  should simply be substituted with  $2m$ , where  $m$  is the original length of the diary, as there are now 2 bits for each character:

$$\begin{aligned} & 3(2m)^5 + 5(2m)^3 + 1 \\ & (3 \times 32)m^5 + (5 \times 8)m^3 + 1 \\ & 96m^5 + 40m^3 + 1 \end{aligned}$$