COMP1201 Tutorial 3

Problem Solving Questions

Due: 4pm 17/05/2019

This tutorial contains two algorithmic problem solving questions. They are different from the previous tutorials in the way that they do not require implementation. Instead, their focus is on the analytic problem solving skills of the module. Please justify your answers with **detailed** explanations. We **will not accept answers without explanations**, even if they are correct.

Note that there are hints for each question. Therefore, if you use those hints and answer them first, tthey will eventually lead to the final solution of the question (answering those hints will also help you writing down the answer in a structured way).

1. [*5 marks*] Consider the following hash function with separate chaining: It takes a non-negative integer $n < 100000$ and calculates its hash value as $(2d_1 + 3d_2 + 5d_3 + 7d_4 + 11d_5)\%47$, where % is modulo devision, d_1 is the most significant digit, and d_5 is the least significant one (the others are therefore in decreasing significance order). Suppose we have 2000 numbers. Prove that there exists a positive integer x for which we can find at least 43 numbers among the given 2000, whose hash value equals to x.

   Some hints to solve this problem:

   - Can you check how many different hash values can be?
   - Now, suppose that the statement is incorrect. Maximum how many numbers the hash table can store?

2. [*5 marks*] Rick, the mad scientist has developed a quantum text compression algorithm, which can take any binary sequence of length $n > 0$ and compress it into a single memory slot of his computer. The required running time complexity for this process is $3n^5 + 5n^3 + 1$. Morty, Rick's grandson, wants to test this algorithm on his private diary, but only to realise that the algorithm cannot read the text as his diary was written as a sequence of three characters (circles, squares, and triangles). Can you help Morty to modify the compression process so that it can work on his diary? Note that as Morty does not know how the original algorithm was implemented, so he cannot directly modify the algorithm. Thus, he can only use it as a black-box tool. How much running time complexity the new algorithm will need? Please give the time complexity as a function of m, the length of Morty's original diary.

   Some hints to solve this problem:

   - Think about how to modify the text of Morty's diary to make it compatible with the compression algorithm.

- How much increase this will cause to the length of the text?

3. **Bonus question, no marks for solving this, but I will give some nice chocolates for those who correctly solve this one. You do not have to submit this question with the other two, but send its solution in a pdf form to Long Tran-Thanh: l.tran-thanh@soton.ac.uk. Please indicate in the subject line that it is for the bonus question of Tutorial 3.**
   Given n distinct positive integers $x_1, x_2, \ldots, x_n$. We would like to find the largest subset of these numbers such that the elements of this subset form an arithmetic sequence with difference of 2019 (i.e., the subset forms a sequence of $\{x, x+2019, x+4038, x+6057, \ldots\}$). Propose a solution that only requires $O(n \log n)$ operations. Please explain your answer.