OLLSCOIL NA hÉIREANN
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

2017/2018

Semester 1 - Winter 2017

CS3509: Theory of Computation

Professor Omer Rana (Extern)
Professor Cormac Sreenan
Professor Michel Schellekens

One and a half hours

Total marks: 80

Answer all Questions

Calculators allowed

PLEASE DO NOT TURN THIS PAGE UNTIL INSTRUCTED TO DO SO

ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER

Question 1 [20 marks]
a) [3 marks] State the Kraft inequality.

b) [3 marks] Under which condition is the Kraft inequality an equality?

c) [4 marks] Compute the entropy of a biased 8-sided dice, where the outcomes of the dice have the following probabilities:

$$prob(1) = prob(2) = \ldots = prob(6) = prob(7) = 0.12$$

while

$$prob(8) = 0.16$$

d) [2 marks] If a comparison-based sorting algorithm has $O(nlog_2(n))$ worst-case time, what can you conclude about its average-case time? (No need to prove the result).

e) [4 marks] What are the four main properties of comparison-based algorithms? Give their names and state briefly for each property what they mean (no proof required, merely a clear explanation of what each property means).

f) [4 marks] Which of these codes cannot be Huffman codes for any probability assignment? If you decide that the code cannot be a Huffman code, then explain why this cannot be the case. If you decide that it is a Huffman code, then give a probability assignment and alphabet that would lead to this code.

1. 0, 10, 11 [2 marks]
2. 00, 01, 10 [2 marks]

Question 2 [20 marks]
a) [15 marks] Can you write a program that checks for any given Python program P consisting of a while loop [while B do Q] whether the program P will ever execute the body Q of the while loop? If so, provide such a program. If not, provide a proof that the program cannot exist.

b) [5 marks] Is it possible for a comparison-based sorting algorithm to sort $\frac{1}{n^2}$ of its inputs of size n in linear worst-case comparison-time? Here, we assume that on the other inputs the algorithm is allowed to run arbitrarily slower. Justify your answer.

Question 3 [20 marks]
We present a variant of Sequential Search which searches for two targets: $target_1$ and $target_2$. We assume that the two targets are distinct, i.e., $t_1 \neq t_2$. We also assume that the targets must occur in any list we search. As usual, for our analysis, we will assume that lists consist of pairwise distinct elements. Finally we assume that the OR in the code is not a lazy operation, i.e., both parts of the OR are evaluated and two comparisons are made per execution of the OR statement.

This version of Sequential Search runs through the list in left to right order, comparing each element to both targets and records the position of a match. As soon as it finds the positions

i_1 and j_1 for both targets t_1 and t_2 it returns these positions as a pair (i_1, i_2).

The pseudo-code for Sequential-Search for two targets, or SS2, is given by:

Sequential Search$(\mathbf{L}, \mathbf{target_1}, \mathbf{target_2}, \mathbf{n})$
$M = \emptyset; halt = 0$
  $for\ i = 1\ to\ n\ do$
    $if\ halt\ =\ 2\ then\ return\ M$
    $if(L[i]\ =\ target_1\ or\ L[i]\ =\ target_2)$
      $then\ append(M, (i));\ halt\ :=\ halt + 1$

Determine the worst-case comparison time [2 marks] and the average-case comparison time for SS2 [18 marks].

Hint 1: For the average case analysis, you can start your argument as follows:

We consider the case where the first occurrence of a target is in position k and the second occurrence of a target is in position l where $1 \leq k < l \leq n$.
The situation is as follows:

$$L = (L[1], \ldots, \mathbf{L[k]}, \ldots, \mathbf{L[l]}, \ldots, L[n])$$

where:

$$(L[k] = t_1 \text{ and } L[l] = t_2) \text{ or } (L[k] = t_2 \text{ and } L[l] = t_1)$$

Carry out the analysis for the case where $L[k] = t_1$ and $L[l] = t_2$. Note that for the converse case, where $L[k] = t_2$ and $L[l] = t_1$ the same number of comparisons will be made. So at the end of the argument, we double up the total number of comparisons to ensure that both cases have been taken into account.

Hint 2: In your computations you can use the following facts without proof:

$$\sum_{l=2}^{n} l(l-1) = \sum_{l=1}^{n-1} l(l+1)$$

$$\sum_{l=1}^{n-1} l(l+1) = \frac{(n+1)n(n-1)}{6}$$

Question 4 [20 marks]
a) [4 marks] Explain how Gödel encoding encodes a program into a natural number.
b) [5 marks] Show that for any programming language there are a countable number of programs.
c) [8 marks] Provide the detailed diagonalization argument used to show that there are uncountably many binary sequences of infinite length.
d) [3 marks] Recall that a decision function is a function with domain the natural numbers and co-domain the values 0 and 1. Demonstrate (making use of the facts given in Questions 4 b and 4c) that there must exist decision functions that are not computable.