S228/102

DUBLIN INSTITUTE OF TECHNOLOGY
KEVIN STREET DUBLIN 8

BSc. (Honours) Degree in Computer Science

PROG 1210

Year 1

SEMESTER II EXAMINATIONS 2014

ALGORITHM DESIGN & PROBLEM SOLVING

A. Curley
Dr. D. Lillis

Duration: 2 hours

Friday 16th May, 4.00-6.00

Answer question (1) and ***any two*** of questions (2), (3), (4)

Question (1) is worth **40** marks.
Questions (2), (3), (4) are worth **30** marks each

1. (a) **(i)** Write a recursive algorithm for adding up a list of numbers.

(8 marks)

(ii) Illustrate the call stack, when the algorithm in 1(a)(i) is implemented, using [9, 5, 7, 4, 10, 11].

(6 marks)

(iii) What is the complexity of the algorithm in 1(a)(i)? Explain your answer.

(2 marks)

(b) **(i)** Describe briefly how the *insertion sort* algorithm works.

(4 marks)

(ii) Rewrite the insertion sort algorithm to sort non-increasing instead of non-decreasing order.

(6 marks)

(iii) What is the complexity of the algorithm described in 1(b)(i)? Explain your answer.

(2 marks)

(c) Write a Haskell function findMax that calculates the minimum of a list of integers. Then show how findMax [5, 3, 1, 7, 8] would be evaluated.

(8 marks)

(d) Express the function $\boldsymbol{n^3/1000-100n^2-100n+3}$ in terms of Big O notation.

(4 marks)

2. (a) i. What is a *binary search,* and in what scenario(s) is a binary search best used?

(6 marks)

ii. Write an algorithm to find the **eight of hearts** in a pack of playing cards. The cards are sorted and the order of suits is diamonds, hearts, spades and clubs.

(10 marks)

(b) i. Write a recursive algorithm (in pseudo code), which provides a solution to the *Tower of Hanoi* problem.

(10 marks)

ii. What is the complexity of the algorithm in 2(b)(i)? Explain your answer.

(4 marks)

3. (a) i. Describe what the *Sieve of Erastosthenes* algorithm does.

(3 marks)

ii. Write, in pseudo code, the Sieve of Erastosthenes algorithm.

(6 marks)

(b) i. Describe the *eight puzzle.*

(4 marks)

ii. Show some of the state space (at least 5 states) for the eight puzzle, starting from any state of the puzzle.

(7 marks)

(c) i. Write a function, in pseudo code, for calculating income tax. There are two tax rates, which are set to 22% and 40% respectively. There is also a tax free allowance, and any income up to this amount is not taxed. There is also another value - *limit,* and any income between the tax free allowance and limit is taxed at 22%. Income over limit is taxed at 40%.

(5 marks)

ii. Write the function in 3(c)(i) in Haskell.

(5 marks)

4. (a) Describe briefly how the *bubble sort* algorithm works.

(6 marks)

(b) Illustrate how the bubble sort algorithm works on the list: [3, 1, 4, 7, 5, 9, 6, 2].

(6 marks)

(c) Modify the bubble sort algorithm so that the performance of this algorithm over an already-sorted list is O(n).

(9 marks)

(d) Illustrate how the modified bubble sort algorithm in 4(c) works on the list: [1, 2, 3, 4, 5, 6, 7, 8].

(3 marks)

(e) Briefly compare the performance of merge sort with that of bubble sort for a list of size 1024.

(6 marks)