S228/102

**DUBLIN INSTITUTE OF TECHNOLOGY**

**KEVIN STREET DUBLIN 8**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#### BSc. (Honours) Degree in Computer Science

#### 

**Year 1**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## SEMESTER II EXAMINATIONS 2016

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

##### ALGORITHM DESIGN & PROBLEM SOLVING

A. Curley

Dr. D. Lillis

Duration: 2 hours

Friday 13th 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)** | 1. Write the pseudo-code for the *Towers of Hanoi* algorithm.   (8 marks)   1. What are the first eight calls placed on the call stack when the Tower of Hanoi algorithm is implemented with four disks.   (12 marks)   1. What is the complexity (Big O) of the algorithm in 1(a)(i)? Explain your answer.   (4 marks) | |  |
|  | **(b)** |  | 1. Write a recursive algorithm for adding up a list of numbers.   (8 marks)   1. Illustrate the call stack, when the algorithm in 1(b)(i) is implemented, using [8, 3, 7, 4, 2, 1].   (6 marks)   1. What is the complexity of the algorithm in 1(b)(i)? Explain your answer.   (2 marks) | |
|  |  |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **2.** | **(a)**  **(b)** | 1. Describe what the *Sieve of Erastosthenes* algorithm does.   (3 marks)   1. Write, in pseudo code, the *Sieve of Erastosthenes* algorithm.   (5 marks)   1. The following algorithm is an incorrect version of the *Selection Sort*. Using the following list highlight exactly where the issues are – [2, 6, 1, 3].   (8 marks)  for i = 0 to i < N-1 do  min = A[i]  for j = i to j < N do  if A[j] < A[min] then  min = A[j]  temp = A[j]  A[min] = A[i]  A[i] = temp  End for   1. Which sorting algorithm would be best suited to sort the following list – [1, 3, 2, 6]? Why have you chosen this algorithm?   (8 marks)   1. Draw a flowchart for the sorting algorithm that you chose in 2(b)(ii).   (6 marks) | (5 marks) |

|  |  |  |  |
| --- | --- | --- | --- |
| **3.** | **(a)**  **(b)** | 1. Write an algorithm, in pseudo code, for the recursive *binary search* algorithm.   (8 marks)   1. Illustrate how your algorithm in 3(a)(i) works, using a **call stack**, when the element you are searching for is 11 in the following list – [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].   (6 marks)   1. What is the complexity of the algorithm in 3(a)(i)? Explain your answer.   (4 marks)     1. Draw a flowchart of an iterative version of *Factorial*.   (6 marks)   1. Write a Haskell function *fact* that calculates the Factorial series of an integer.   (6 marks) |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **4.** | **(a)** | The *Merge Sort* algorithm is described in pseudo code below.   1. What issues are there with the incorrect Merge Sort Algorithm below?   (9 marks)  ***Merge\_Sort (A, low, high)***  *If n =< 2*  *Return*  *Else*  *mid = (low+high)/2*  *Merge\_Sort(A, low, high)*  *Merge(A, low, mid, high)*  *Merge\_Sort(A, mid+1, high)*   1. Describe the *Merge* part of the Merge Sort algorithm in pseudo code.   (8 marks)   1. Illustrate how the following lists can be merged using the algorithm described in 4(a)(ii) – [1, 6, 7,8] and [2, 3, 4, 5].   (4 marks)   1. What are the first **nine** function calls made on a list of eight elements? The Merge function should be included where required.   (9 marks) | (5 marks) |