THE UNIVERSITY OF THE SOUTH PACIFIC

**CS214: Design and Analysis of Algorithms**
School of Computing, Information and Mathematical Sciences

Final Examination
Semester 2 2015

Mode: Face to Face

**Duration of Exam: 3 hours + 10 minutes**

Reading Time: 10 minutes

Writing Time: 3 hours

**Instructions:**

1. This is a closed book exam
2. This exam has six sections – I, II, III, IV, V and VI
   a. Section I: 6 Marks
   b. Section II: 4 Marks
   c. Section III: 10 Marks
   d. Section IV: 10 Marks
   e. Section V: 4 Marks
   f. Section VI: 6 Marks
3. All questions are compulsory.
4. Write your answers in this booklet.
5. This exam is worth 40% of your overall mark.
6. Number of pages (including this cover page): 18

**Name:** ________________________

**ID:** ________________________

**Section I: Multiple Choice** [6 Marks]

Circle the letter of the correct answer for Questions 1-6 in the grid provided below. Each question can have more than one correct answer.

| Question | Answer | | | | | |
|---|---|---|---|---|---|---|
| 1 | a | b | c | d | e | f |
| 2 | a | b | c | d | e | f |
| 3 | a | b | c | d | e | f |
| 4 | a | b | c | d | e | f |
| 5 | a | b | c | d | e | f |
| 6 | a | b | c | d | e | f |

1. A problem can be solved by using either a Divide-and-Conquer approach or a Dynamic Programming approach. Which approach is better?
    a. Divide-and-Conquer approach is better as it divides the problem into half size to reduce the time complexity
    b. Dynamic Programming approach is better as it is dynamic
    c. Dynamic Programming approach is better as it does not use recursive formula
    d. Dynamic Programming approach is better as it applies the Principle of Optimality
    e. Dynamic Programming approach uses arrays but Divide-and-Conquer approach does not use any array
    f. Depends on the requirements and available resources

2. Which of the following statements is/are true for Minimum Spanning Trees?
    a. A Minimum Spanning Tree is a shortest path
    b. A Minimum Spanning Tree is a tree where all nodes are directly connected
    c. Minimum Spanning Trees can be applied in designing networks such as electrical grids, TV cables, computer networks, and water supply networks
    d. A solution to construct a Minimum Spanning Tree is also a solution to a Travelling Salesperson Problem
    e. All of the above
    f. None of the above

3. Big O, Omega and Order
    a. Big O concerns only with eventual behaviour. It describes the asymptotic behaviour of a function
    b. Big O puts an asymptotic lower bound on a function so it is used to describe worst case time complexity
    c. Omega puts an asymptotic upper bound on a function so it is used to describe best case time complexity
    d. Big O is at least as good as a pure quadratic function, whereas Omega is at least as bad as a pure quadratic function.
    e. Order is a combination of Big O and Omega
    f. All of the above

4. Consider the following three sets of binary codewords:

   Set A {01, 10, 011, 010}

   Set B {01, 10, 001, 111}

   Set C {01, 10, 001, 1100}

   a. Set A is more efficient as the length of bits is shorter than Sets B and C
   b. Set B is more efficient as there are only 4 bits of "0"
   c. Set C is more efficient as there are less number of bits of "1"
   d. Cannot compare their efficiency as not all the sets are prefix-free
   e. Set B is more efficient as its length of bits is less than Set C
   f. None of the above

5. Which of the following algorithms use greedy approach?
   a. Dijsktra's Algorithm, Floyd's Algorithm and Kruskal's algorithm
   b. Dijsktra's Algorithm, Kruskal's algorithm and Prim's algorithm
   c. Kruskal's algorithm, Huffman's Algorithm and Prim's algorithm
   d. Floyd's algorithm and Huffman's Algorithm
   e. All of the above
   f. None of the above

6. An algorithm has an every case time complexity if
   a. Its best case time complexity, average time complexity and worst case time complexity are the same
   b. The basic operation gets excused once only, it will never have an every case time complexity
   c. The basic operation is always done the same number of times for every instance of the input size
   d. Mergesort has an every case time complexity
   e. Quicksort does not have an every case time complexity
   f. None of the above

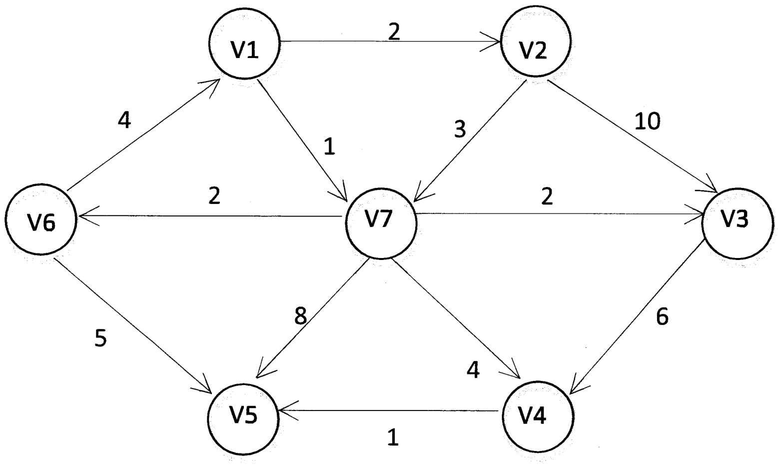
**Section II: Huffman's Algorithm** [4 marks]

Use Huffman's Algorithm to construct two codes that can be used to encode the following string. Ignore the space. Show if the two codes you constructed have the same efficiency for the purpose of data compression.

"I saw the saw saw a saw"

**Section III: Shortest Path** [3+3+4 Marks]

Given the following graph and a set of algorithms namely Dijsktra's Algorithm, Floyd's Algorithm, Kruskal's algorithm, Huffman's Algorithm and Prim's algorithm




1. Choose an algorithm that applies the Principle of Optimality to compute the shortest path from V6 to V3. Show the computation steps. Show the resulting path. Explain how the Principle of Optimality applies in the chosen algorithm.

*(continue your answer to question 1 on this page, if need)*

2. Choose an algorithm that implements a greedy approach to compute the shortest path from V6 to V3. Explain what the algorithm is greedy on.

3. Compare the major similarity and difference between the two chosen algorithms.

**Section IV: The 0-1 Knapsack Problem** [3+(4+3) Marks]

1. For $i > 0$ and $w > 0$, let $P[i][w]$ be the optimal profit obtained when choosing items only from the first $i$ items, such that the total weight less than or equal to $w$. Then,

$$P[i][w] = \begin{cases} P[i-1][w] & , if\ w_i > w \\ maximum(P[i-1][w], p_i + P[i-1][w-w_i]), & if\ w \geq w_i \end{cases}$$

can be used to describe three different cases and compute an optimal solution to the 0-1 Knapsack Problem. Give three examples that can show three different cases with the $P[i][w]$ formula (above). Each example has at least 3 items. Use the $P[i][w]$ to represent the 0-1 Knapsack Problem of the given examples.

2. Consider the following 0-1 Knapsack Problem:
   item 1: $80, 20kg
   item 2: $12, 2kg
   item 3: $50, 10kg
   knapsack maximum capacity: 16kg
   1) Use a Best-First Search with Branch-and-Bound approach to show a pruned state space tree. Show the node from which an optimal solution can be found. Show the sequence of the nodes visited.

2) Compare this approach with the Backtracking approach in constructing a state space tree for 0-1 Knapsack Problem. Are the comparison results applicable to problems other than the 0-1 Knapsack Problem?

**Section V: Minimum Spanning Tree** [1+3 Marks]

1. Can a graph have more than one minimum spanning tree? Give an example to illustrate your answer.

2. Choose an algorithm from the set of algorithms namely, Dijsktra's Algorithm, Floyd's Algorithm, Kruskal's algorithm, Huffman's Algorithm and Prim's algorithm, to construct minimum spanning tree(s) from the example you gave above to justify your answer to 1.

**Section VI: Algorithm Design** [3+3 Marks]

Design an algorithm for the following scenario

> *A medical specialist doctor sees 5 patients the first Tuesday of every month. He requires his assistant to schedule one appointment for each patient. Since bookings are required to be made at least one month in advance, the assistant does not need to consider the booking time when scheduling the appointments but simply assign one patient to a timeslot.*

1. How many ways the assistant can schedule the appointments in terms of who sees the doctor in which timeslot? Write an algorithm that can provide an answer to the question above. The algorithm will have an $\Theta(n)$ time complexity. Your algorithm must be presented in C++ or C++ like pseudocode.

2. Show if your algorithm also have an $O(n^2)$ time complexity.

**Extra Working Page 1**

**Extra Working Page 2**

**Extra Working Page 3**

THE END