School of Computing and Information Technology

Student to complete:

Family name	
Other names	
Student number	
Table number	

CSCI203
Algorithms and Data Structures
Wollongong and SWS Campus

Examination Paper
Spring Session 2021

Exam duration	3 hours
Weighting	55%
Items permitted by examiner	Open Book
Aids supplied	None
Directions to students	8 questions to be answered. The marks of each question is printed alongside each question. • Answer each question on a separate page clearly • Convert the answers into **one** pdf file • Submit the pdf file.

Question 1: (8 marks)

All parts of this question relate to the following array which represents a min-heap:

5	**14**	**23**	**32**	**41**	**87**	**90**	**50**	**64**	**53**	**43**	

a) Draw the binary tree corresponding to this array.

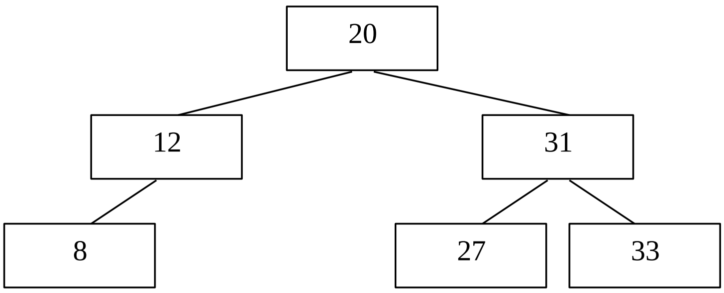
b) Show the array that results when the first element with value 5 is removed and the heap is rebuilt. Show your working.

c) Show the result when an element with value 16 is added at the end of the original array and the heap is rebuilt. Show your working.

d) What is the complexity of adding an element to a heap?

Question 2: (8 marks)

All parts of this question relate to the following AVL tree:



a) Show the tree after inserting a new node with value 9 into the tree before it is rebalanced.

b) This insertion causes the tree to become unbalanced. At which node does the imbalance occur?

c) Show the tree after rebalancing to restore the AVL criterion.

d) Insert a new node with value 29 into the tree obtained after insertion of the node with value 9, show the tree before and after rebalancing if needed.

Question 3: (6 marks)

Both parts of this question relate to the following array:

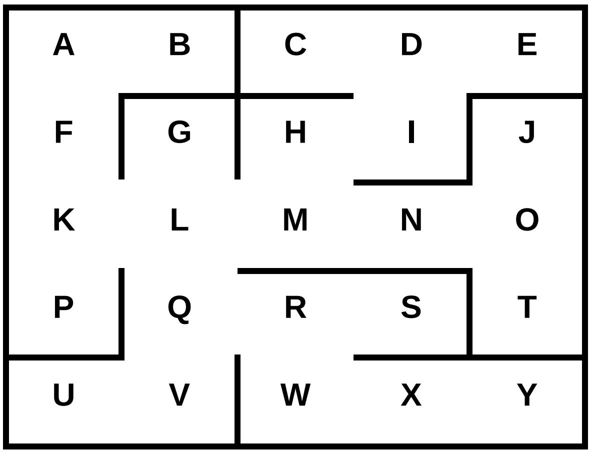
57	**44**	**27**	**81**	**45**	**39**	**64**	**42**	**17**	**29**

a) Show the resulting array at each stage of a merge sort.

b) After the array has been sorted, how may data comparisons are required to find the number 45 with (i) a linear search, (ii) a binary search.

Question 4: (8 marks)

Both parts of this question refer to the following maze where nodes are visited starting from node **A** and searching for node **Y**. Note: If there are multiple paths out of a given node they should be tried in the order Down, Up, Left and Right.



a) List the nodes in the order in which they are encountered if the maze is traversed using Depth-First search. Which data structure is appropriate for storing nodes that are still to be searched?

b) List the nodes in the order in which they are encountered if the maze is traversed using Breadth-First search. Which data structure is appropriate for storing nodes that are still to be searched?

Question 5: (6 marks)

Explain briefly how the following processes work:

a) Collision resolution using chaining

b) Collision resolution using open address hashing.

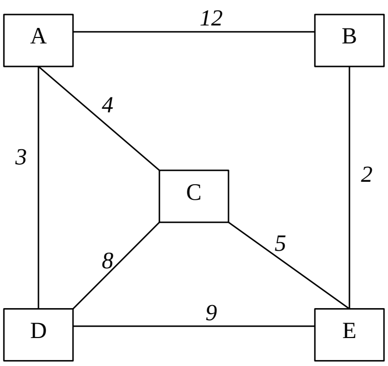
c) Consider inserting the keys $10, 22, 31, 4, 15, 28, 17, 88, 59$ into a hash table of length $m = 11$. Show the result of inserting these keys using a linear probing with $p(k, i) = (h(k) + i)\ mod\ (m)$, where k is a key and i is an iteration count starting from 0

Question 6: (6 marks)

Both parts of this question relate to the graph to the right.

a) Write the adjacency matrix for the graph.

b) Using Dijkstra's Algorithm, determine the minimum distance from node A to each other node. Show your working.



Question 7: (6 marks)

a) Draw an appropriate expression tree for representing the following mathematical equation:

((x * x) + 2) / (x + 1)

b) Rewrite the expression in postfix notation (also known as post-order or reverse Polish notation).

c) Using diagrams, show how the following postfix expression would be evaluated by using a stack.

x y + y z w * - *

Show each stack operation in your answer.

Question 8: (7 marks)

a) Briefly explain the process of developing an algorithm using dynamic programming technique. Use the problem of Fibonacci numbers as an example.

b) Use the bottom-up dynamic programming algorithm (Slide 18, Week11-B) to find an optimal solution, i.e. optimal value and set of selected items, to the following 0-1 Knapsack problem. Show the optimal values for its subproblems, i.e. $B[k, w], k = 1, \cdots, 4; w = 1, \cdots, 5$, the value of the most valuable subset of the first k items that fits into the knapsack of capacity w

$$B[k, w] = \begin{cases} B[k-1, w] & if\ (w - w_k) < 0 \\ max\{B[k-1, w], B[k-1, w - w_k] + b_k & if\ (w - w_k) \geq 0 \end{cases},$$

where b_k is the value of $k'th$ item.

item	weight	value
1	3	$12
2	2	$10
3	1	$20
4	2	$15

Capacity $W = 5$

—END OF EXAM PAPER—