Beijing-Dublin International College

SEMESTER 2 FINAL EXAMINATION - 2023/2024

School of Computer Science

COMP2014J Data Structures and Algorithms 2

Dr. Robert Ross
Assoc. Prof. Neil Hurley
Dr. David Lillis *

Time Allowed: 120 minutes

Instructions for Candidates

Answer any 2 questions. All questions carry equal marks.

Instructions for Invigilators

No special instructions.

Question 1

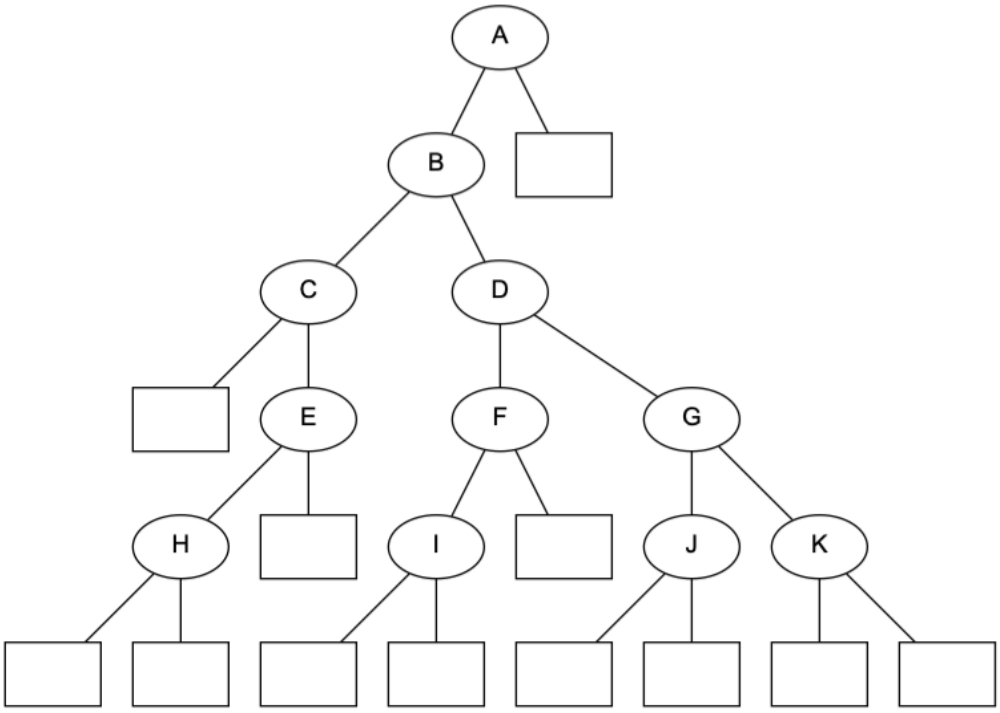


Figure 1(a)

(a) Study the tree in Figure 1(a) and answer the questions that follow.

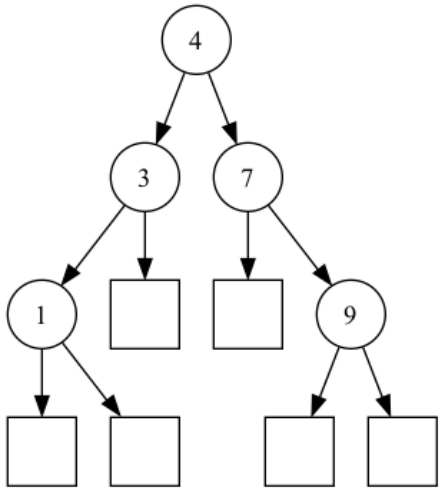
(i) List the siblings of node J.
(ii) What is the depth of node J?
(iii) What is the degree of node A?
(iv) What is the height of the tree?
(v) List the descendants of node C.
(vi) List the ancestors of node E.
(vii) Is (G,K) an edge? Explain your answer.
(viii) List the nodes that are in the subtree that is rooted at G.
(ix) Which of the following phrases is the best description for this tree: *binary search tree, proper binary tree, compete binary tree*. Explain your answer.

[10 marks]

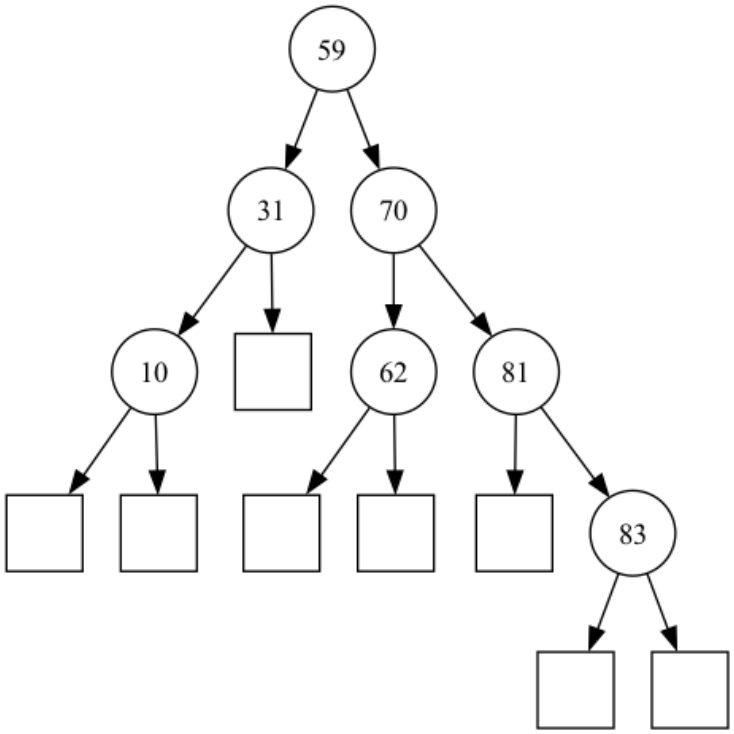
(b) For each of the Figures (i) to (vi) below, assume that the tree is an AVL Tree (where only the keys are shown in the nodes). Draw the state of each tree after performing the operation written above the figure. In your answer, you must also clearly mention any restructuring that is required.

[12 marks]

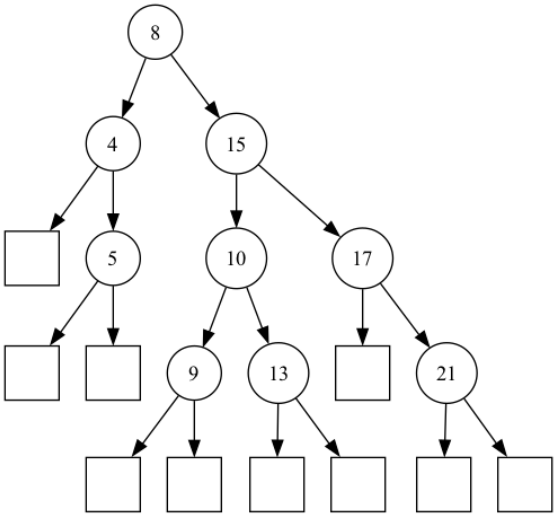
(i) Put key 8



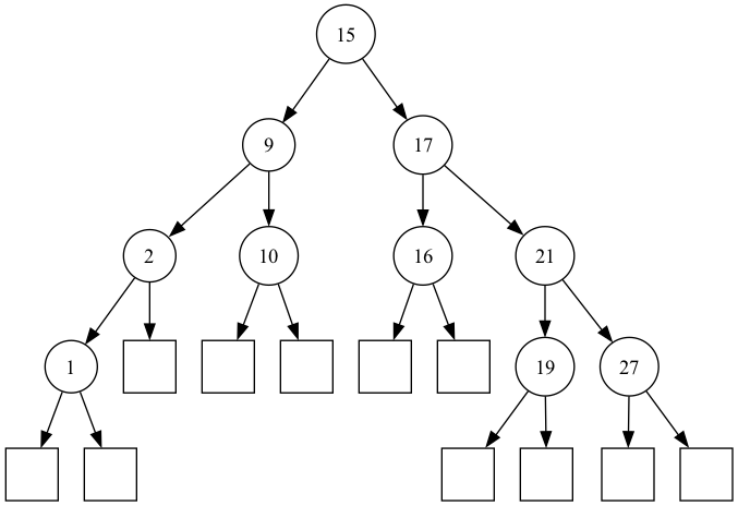
(ii) Remove key 59



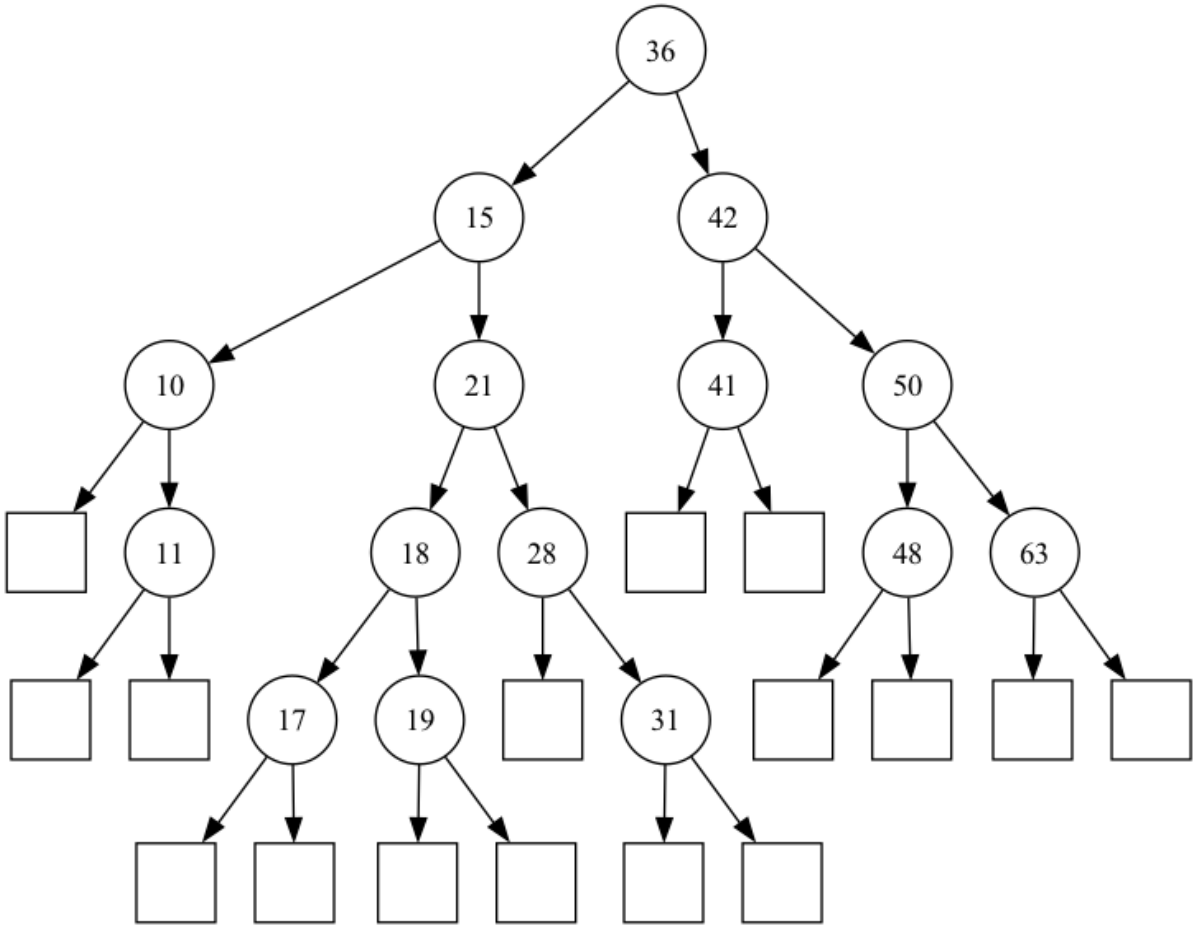
(iii) Put key 12



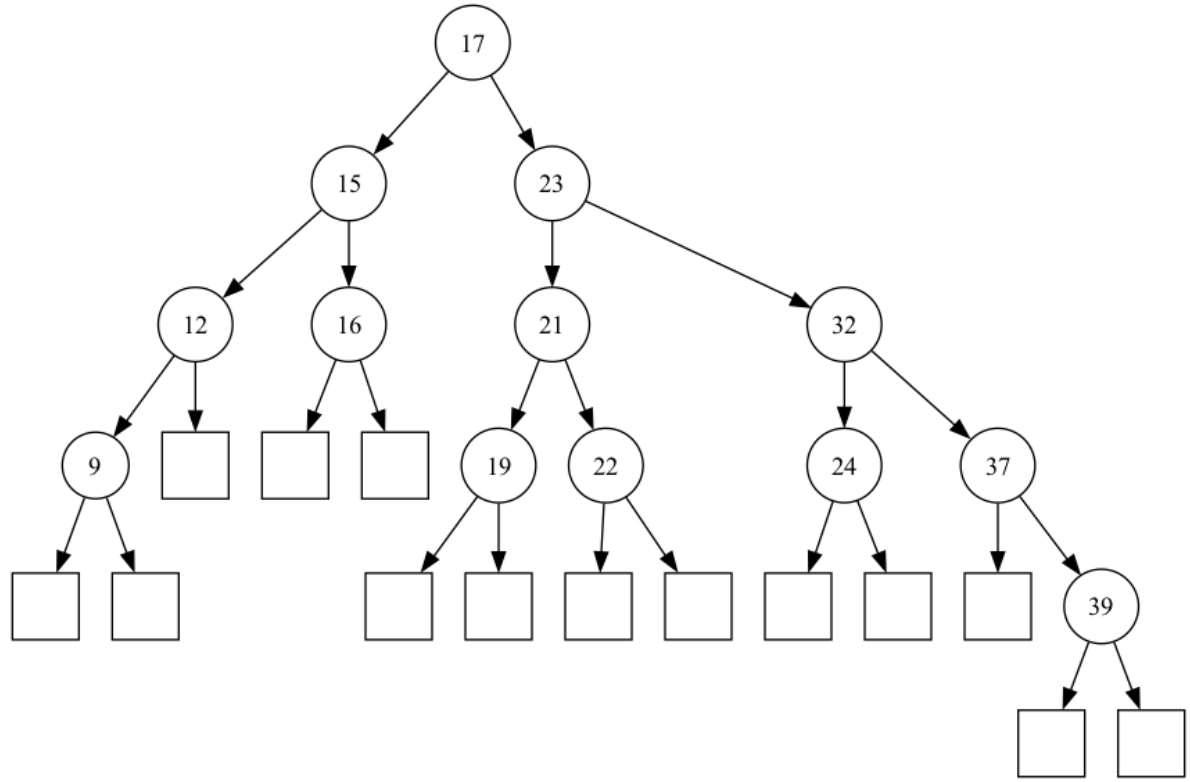
(iv) Put key 26



(v) Remove key 41



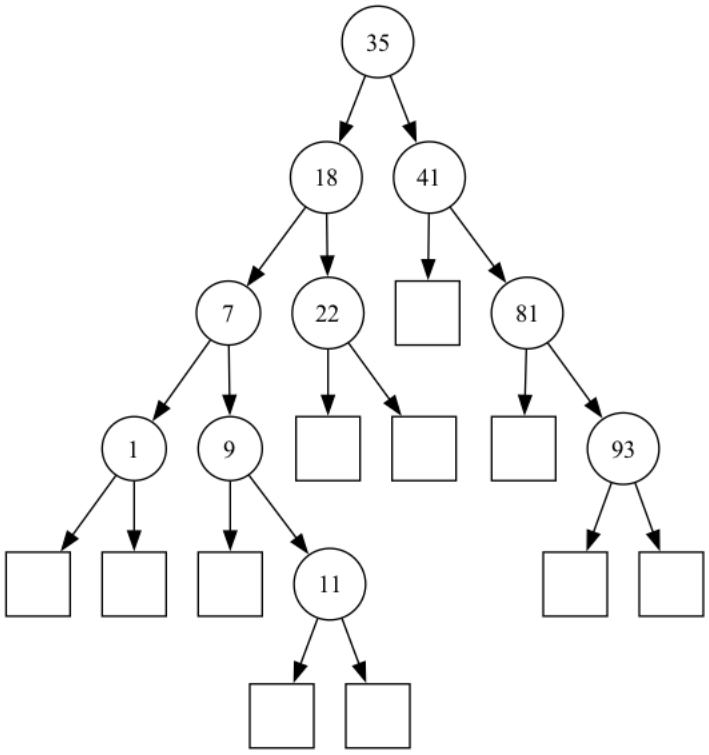
(vi) Remove key 16



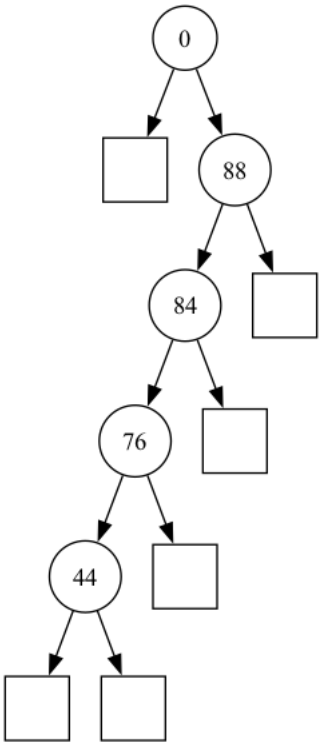
(c) For each of the Figures (i) to (vi) below, assume that the tree is a Splay Tree (where only the keys are shown in the nodes). Draw the state of each tree after performing the operation written above the figure. In your answer, you must also clearly mention any restructuring that is required.

[12 marks]

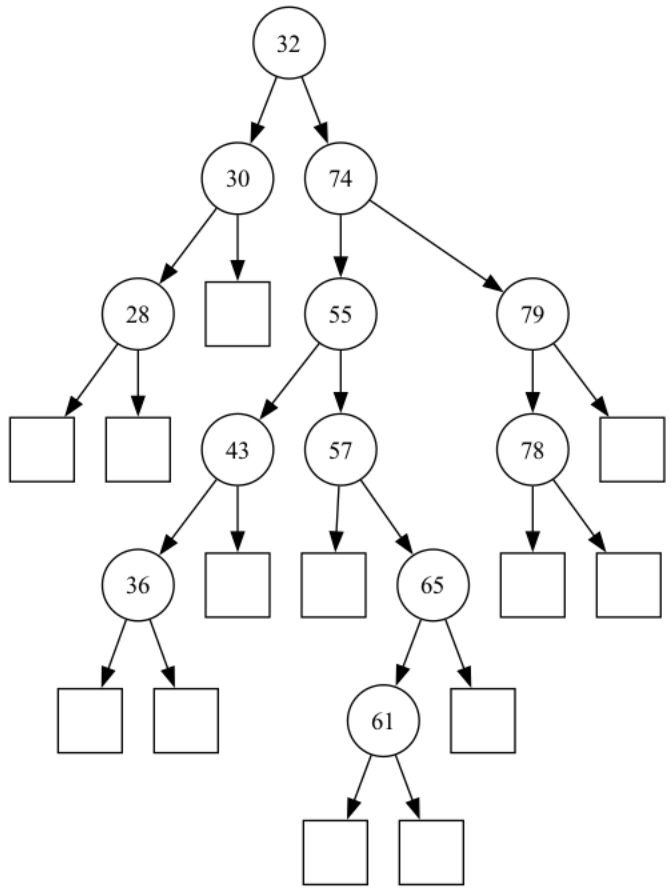
(i) Put key 8



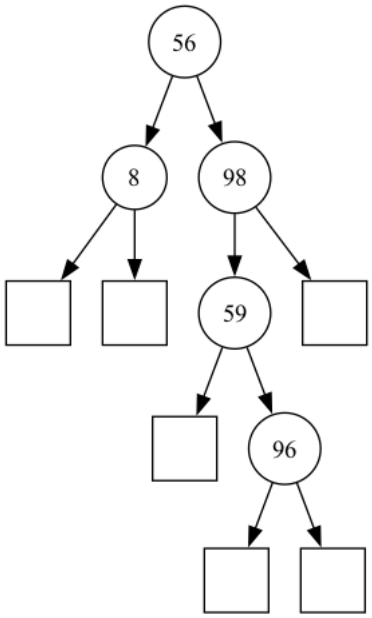
(ii) Remove key 76



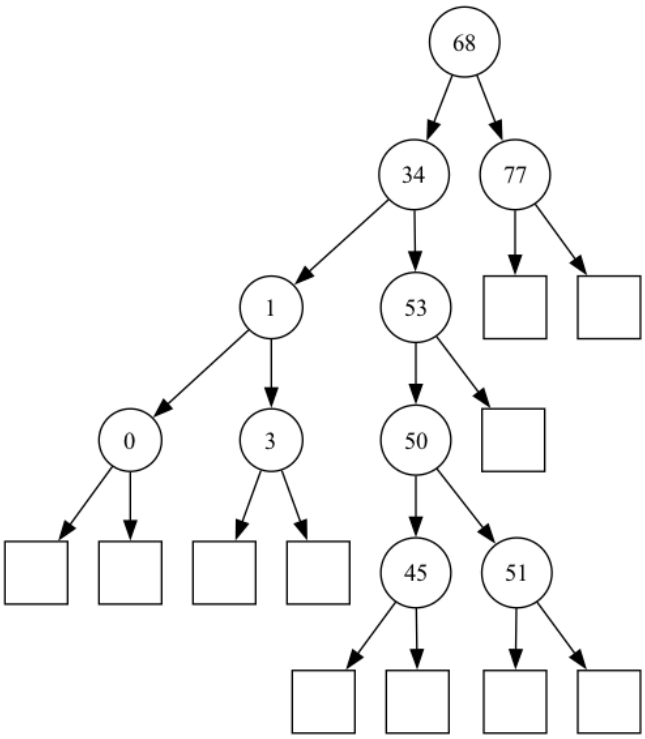
(iii) Put key 66



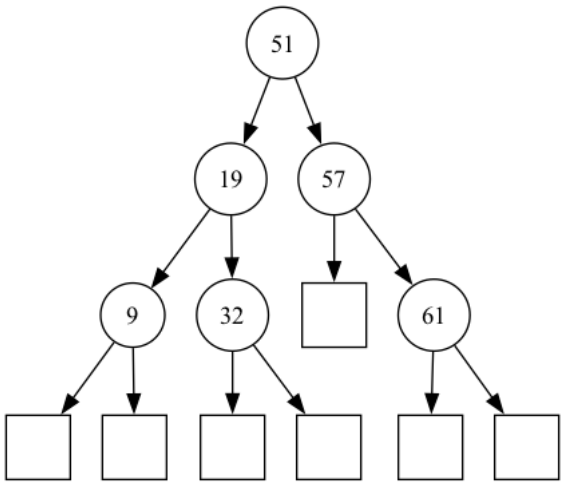
(iv) Remove key 56



(v) Remove key 53



(vi) Put key 2



(d) With reference to an *array-based list*, explain how *amortised analysis* can be used to show that an "insert" operation can be said to run in O(1) time, even though the whole array must be copied when it is full.

[8 marks]

(e) Explain how you would implement a *link-based binary tree*. In your answer, describe the internal structure of a node, identify what key data you must keep track of, and illustrate your description with a diagram of an example tree that contains three nodes.

[8 marks]

[Total 50 marks]

Question 2

(a) Give definitions for the following graph concepts: *connected component*, *incident edges*, *spanning subgraph*, *cycle*.

[8 marks]

(b) Figure 2(b) below represents a Heap-based Priority Queue (only priorities are shown).

(i) Starting from the situation in Figure 2(b), show how the contents of the heap changes after each of the following operations (your answer should show 6 heap diagrams):

insert(50), insert(9), remove(), remove(), remove(), insert(2)

[12 marks]

(ii) In a heap, what is the time complexity of a *downheap* operation? Explain your answer.

[2 marks]

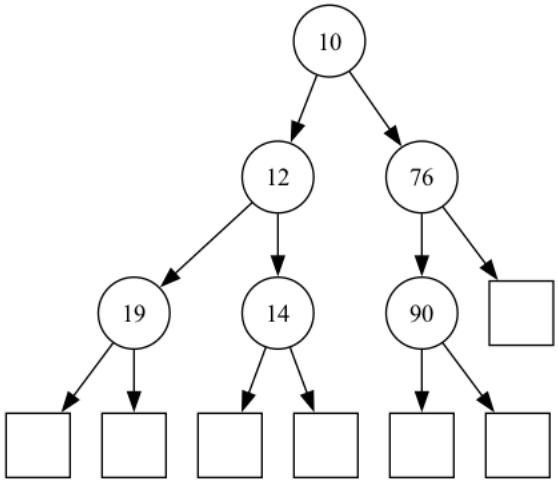


Figure 2(b)

(c) Figure 2(b) above can be described as a Complete Binary Tree (CBT).

(i) Show how the CBT in Figure 2(b) can be represented using an array-based implementation.

[3 marks]

(ii) Explain why an array-based implementation is more suitable than a link-based implementation for a CBT.

[2 marks]

(iii) Briefly explain why an array-based implementation would *not* be suitable for an AVL Tree.

[3 marks]

(d) This part relates to *Huffman Enoding*.

(i) A data file contains 6 unique characters with the following frequencies:

Character	Frequency	ASCII
a	45,000	01100001
b	13,000	01100010
c	12,000	01100011
d	16,000	01100100
e	9,000	01100101
f	5,000	01100110

Using this file as an example, describe *in detail* how a Huffman Tree would be created for this file.

[12 marks]

(ii) Show how your Huffman Tree would be encoded in binary. In your answer, clearly indicate which bits represent the characters and which represent the tree structure.

[4 marks]

(iii) What is the minimum number of bits required to store this file in its compressed format (not including the bits required to represent the Huffman tree)? Explain your answer.

[4 marks]

[Total 50 marks]

Question 3

(a) This question relates to graph traversals.

(i) Give pseudocode for a Depth First Search traversal of an undirected graph.

[5 marks]

(ii) Starting at the vertex containing A, show how a Depth First Search traversal can be conducted for the graph in Figure 3(a). Assume that incident edges are returned in alphabetical order of their opposite vertex.

[9 marks]

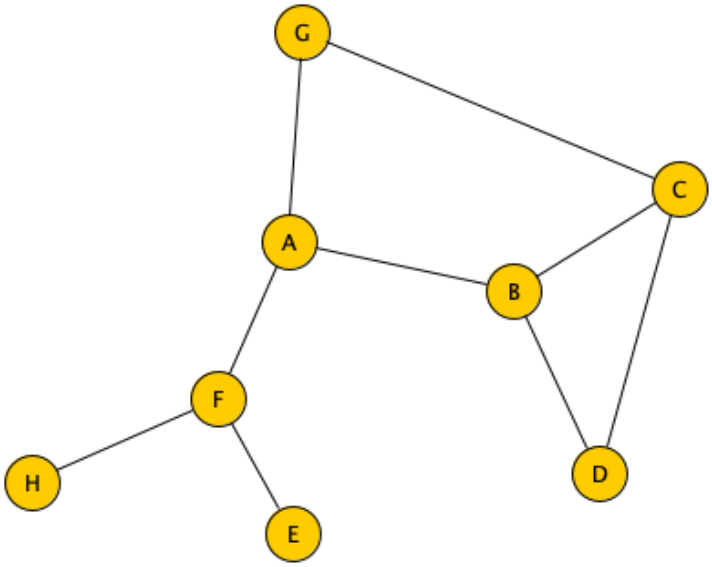


Figure 3(a)

(b) Using the graph in Figure 3(b) below as an example, show how Dijkstra's Algorithm can be used to compute a shortest path tree beginning at A. In your answer, you must explain each step that you take.

[12 marks]

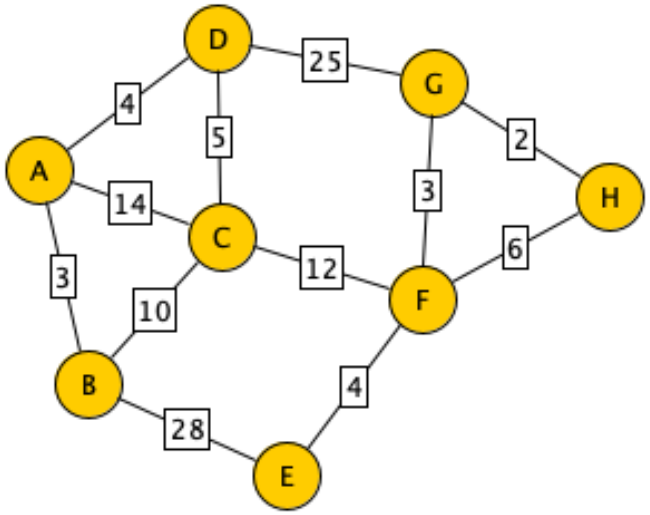


Figure 3(b)

(c) An *adjacency list* structure is one way in which a graph can be implemented. For each of the following graph methods, state what the time complexity is when using an adjacency list structure. You *must* explain the reasons for this complexity in each case.

(i) `incidentEdges(v)`
(ii) `areAdjacent(v,w)`
(iii) `insertEdge(v, w, o)`
(iv) `removeVertex(v)`

[12 marks]

(d) With the aid of a diagram, explain in detail how the graph shown in Figure 3(d) could be represented using an *adjacency matrix* structure.

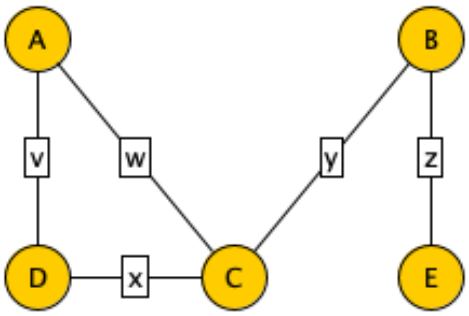


Figure 3(d)

[12 marks]

[Total 50 marks]