

95%

Student ID: 1564710



School of Computing, Electrical and Applied Technology
BCS

ISCG 6426 – Data Structures and Algorithms

Final Exam – Semester 2, 2022

Date: Monday, 21 November 2022

Exam time: 9.00am – 11.10am

Duration: 2 Hours & 10 minutes (including 10 minutes of reading time)

Total Marks: 100

Weighting: 30%

Other Assessment weighting: 70%

No of Questions: 11

Instructions to Candidates:

1. This is an OPEN BOOK exam.
2. You are allowed to bring in and use one A4 double-sided handwritten reference sheet with you. You are not allowed to seek assistance from other people.
3. Write answers in the space provided. If required, you can extend your answers in separate sheets of paper (ask invigilator) and attach it herewith.
4. Use the blank paper at the end of this booklet for rough work. Please keep it attached
5. For coding questions in section B, use C# code or pseudo code. State any assumptions you make.

Summary of paper:

Section	Question Type	Marks
A	8 Short Answer Questions	48
B	3 Coding Questions	52
Total Marks		100

PART A (Short Answer Questions)

QUESTION 1 (Total 6 marks) 6 marks

(a) What are the differences between linear and non-linear data structures? (2 marks) 2

Linear data structure: the elements are organized and accessed in sequence. Each element has a relationship with the others before or after or both, but not with the other elements of the collection.
non-linear: there is a hierarchical relationship between the elements.

(b) Name 4 linear data structures (2 marks) 2

Array, Stack, Queue, Linked-List.

(c) Name 2 non-linear structures (2 marks) 2

Trees, Graphs.

QUESTION 2 (Total 6 marks) 6 marks

There are two empty stacks, s1 and s2. The stacks are objects of the Stack class. The Stack class contains the following methods:

- a. public void Push(int data) { }
- b. public int Pop() { }

What are the contents of s1 and s2 after executing the following code?

```
int a=0;
for (a=0; a<5; a++)
{
    s1.Push(a * 3);
    s1.Push(a + 2);
}
for (a=0; a<2; a++)
{
    s2.Push(s1.Pop() + s1.Pop());
}
```

(a) S1: { 0, 2, 3, 3, 6, 4, 9, 5, 12, 6 } 0,2,3,3,6,4 (3 marks) 3

(b) S2: { 18, 14 } (3 marks) 3

QUESTION 3 (Total 6 marks)**6 marks**

With the help of Big O notation, describe the differences between linear search and binary search algorithms for finding a value in the array.

Linear search has a $O(n)$, search each element from beginning to end, or until value found. (6 marks) **6**

Binary Search has a $O(\log(n))$, divide and conquer, elements to search halves with each iteration.

QUESTION 4 (Total 6 marks)**6 marks**

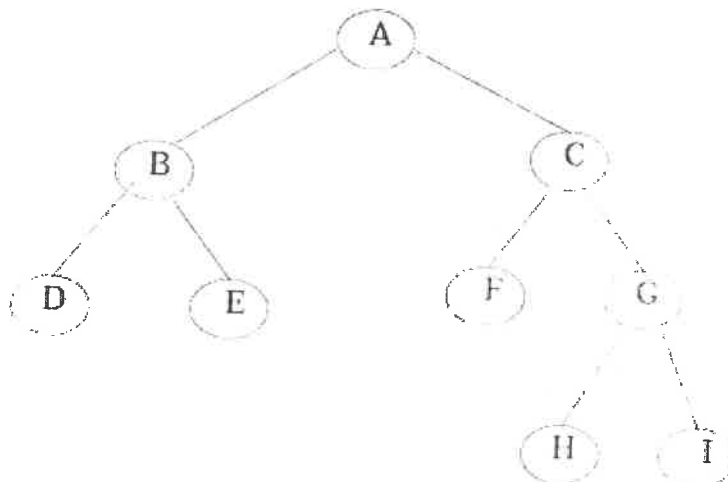
What are differences between Binary Search Trees (BST) and Heaps (Maxheap)?

BST: all values on the left side of the root are small and the right side of the root are large than the root, Standard implementation does not allow duplicate values. A given value can only fit into a specific place on the tree, doesn't need to be complete or nearly complete, can be balanced, data added or removed from anywhere; (6 marks) **6**

Heap: Data organized by value vertically, smaller below, greater above. Data appended to next available leaf, then reheaped-up. ~~Only requirement~~ child values are smaller than parent, must be complete or nearly complete; balance doesn't matter, Removed a node, then reheaped-down.

QUESTION 5 (Total 6 marks)**6 marks**

The diagram displays a binary tree. Show the order in which the data is processed when doing:



(a) Preorder: **A B D E C F G H I**

(b) Inorder: **D B E A F C H G I**

(2 marks) **2**

(c) Postorder: **D E B F H I G C A**

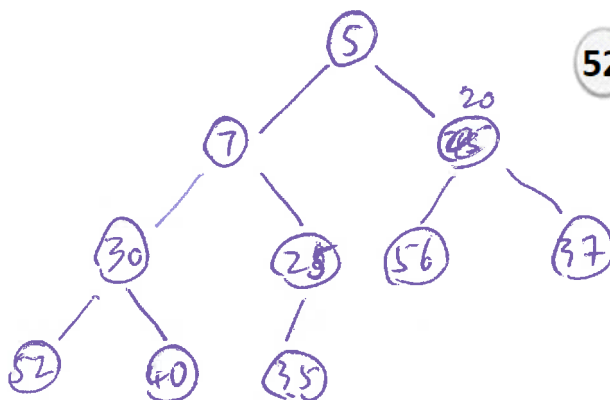
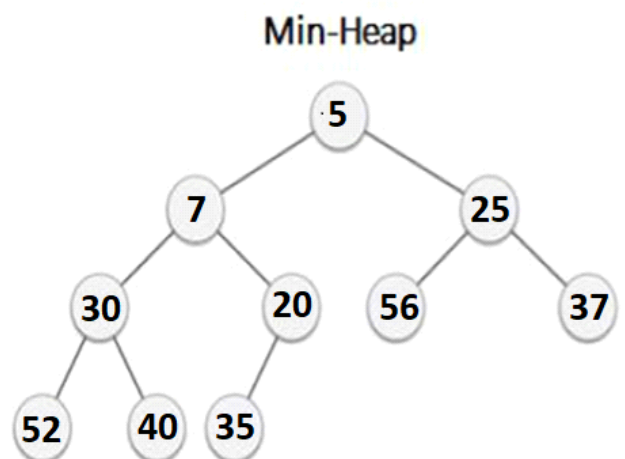
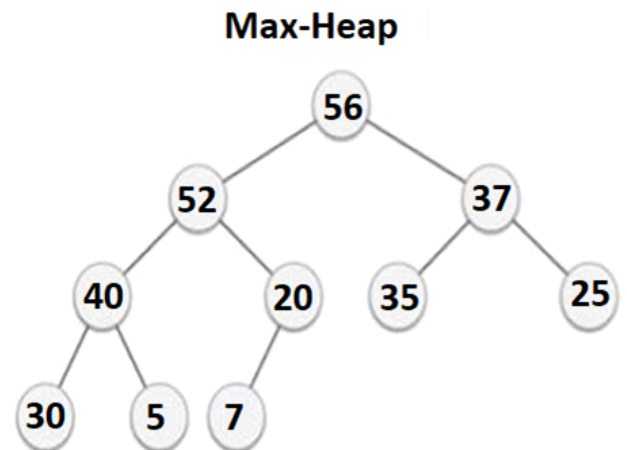
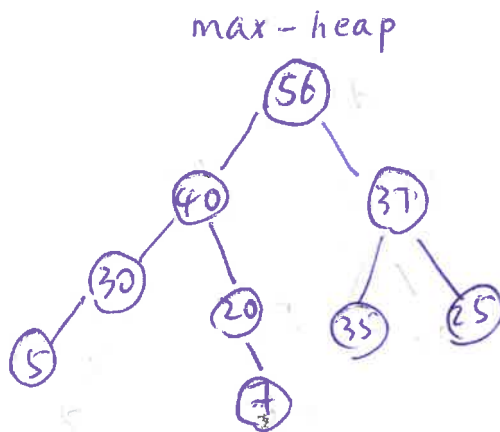
(2 marks) **2**

(2 marks) **2**

QUESTION 6 (Total 6 marks)**3 marks**

Create a node diagram of a max-heap and min-heap for the following input:
30, 35, 56, 40, 7, 37, 25, 52, 5, 20

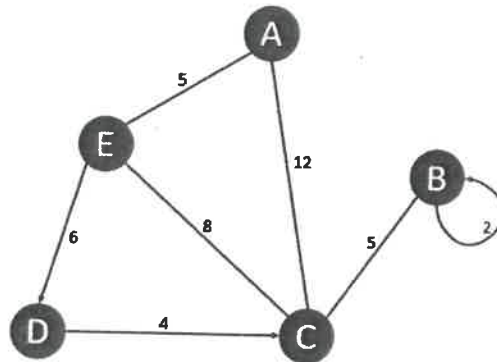
(6 marks) **3**



Your Min-Heap is quite correct except values 20 and 25 need to be swapped. Please revise this topic again.

QUESTION 7 (Total 6 marks)**4 marks**

Examine the diagram below that displays a graph and answer the following questions:



(a) What do the numeric values on each edge describe?

(2 marks) **2**

Edge weight, cost, priority, time.

(b) What differentiates the edge with a value of 2 from the other edges?

(2 marks) **2**

it is a Loop, incident with the same vertex twice.

(c) Starting at vertex A, what is the output of a breadth-first search (list of vertices)?

(2 marks) **0**

~~A, E, D, B~~ ~~A, E, C, D~~

A, E, D, C, B

A, E, C, D, B

Note: For BFT we use a Queue; add the starting vertex to the queue, "deque" to choose the vertex to visit (in this case A), add all neighbours to the Queue (E & C), then "deque" to choose the next vertex to visit (E). Repeat the process by adding all neighbours (D) "A" has been visited, and "C" is already in the queue.

Though the vertex with lower weight is added first, therefore A, E, C, D, B is expected, as the question specifically doesn't say that, if some students considered the value to be "priority" (which is valid) and therefore chose the higher value to add first, A, C, E, B, D can also be accepted.

QUESTION 8 (Total 6 marks)**6 marks**

For the following adjacency list:

A: [C, F]

B: [D]

C: [B, E]

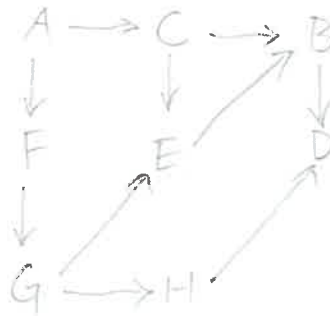
D: []

E: [B]

F: [G]

G: [E, H]

H: [D]



(a) Draw the graph in space above.

(2 marks) **2**

(b) Is this graph a directed-acyclic graph? If yes, why? If no, why not?

(2 marks) **2**

Yes, directed-graph as show the travel direction.

Eg. A to C, but no path from C to A, there is no cycles to return the same vertex.

(c) When does it make more sense to store a graph as an adjacency matrix?

(2 marks) **2**

Adjacency matrix is used where information about each and every possible edge is required for the proper working of an algorithm like: ~~Floyd-Warshall~~ Floyd-Warshall Algorithm where shortest path from each vertex to every other vertex is calculated (if it exists). Sometime it is also used in network flows.

PART B (Coding problems)

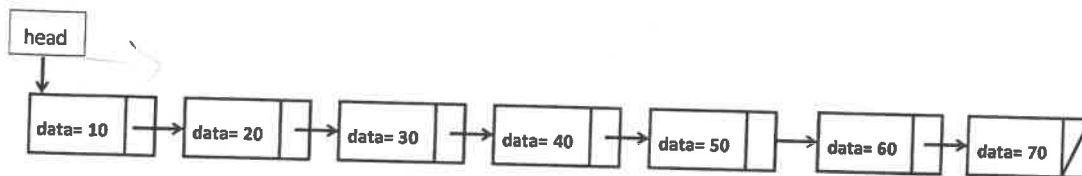
Instructions:

1. Using C# code is preferable, syntactical errors will be excused. You may else use pseudo code
2. Any assumptions or brief explanations can be added as code comments

QUESTION 9 (Total 16 marks)

16 marks

Write code for the Find() method that finds a node with the given value from a linked list. With the following assumptions:



- a. The list is defined as in the figure
- b. The function will have two parameters which are: a reference to the first node of the list, and an integer value to Find.
- c. Returns true if found or false if not found on the list.

```
public int Find(Node root, int Value, int number)
{
    if (head != null)
    {
        Node traversalnode = head;
        while (traversalnode != null)
        {
            if (traversalnode.value != number)
            {
                traversalnode = head.next;
            }
        }
        return true;
    }
    return false;
}
```

QUESTION 10 (Total 16 marks)**16 marks**

Write code for a method that sorts an array into ascending order using Bubble Sort, with the following assumptions:

- Each array element contains a positive integer.
- Return zero if the element is not found.

```
internal int BubbleSort(int[] array)
{
    if (array.Length > 0)
    {
        for (int i = 0; i < array.Length - 1; i++)
        {
            for (int j = 0; j < array.Length - i - 1; j++)
            {
                if (array[j] > array[j + 1])
                {
                    Swap(array, j, j + 1);
                }
            }
        }
    }
    else { return 0; }
}
```


QUESTION 11 (Total 20 marks)**20 marks**

Write code to sort the values of a binary search tree (BST) in ascending order and print to the console.

```
internal void InOrder (BSTNode Root, StringBuilder sb)
{
    if (Root != null)
    {
        InOrder (Root.leftChild, sb);
        sb.Append (Root.data + " ");
        if (Root != null)
        {
            InOrder (Root.rightChild, sb);
        }
    }
    Console.WriteLine (sb);
}
```

Rough work here: