COMP90038
Algorithms and Complexity
Sample Exam 2017*

Leo’s solution

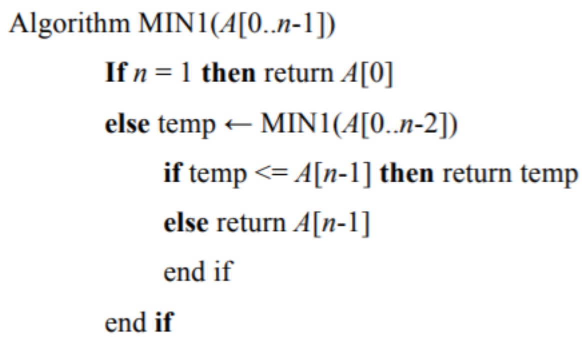
Section A (13 marks): Answer all the questions

1. **(1 mark)** What is the fundamental difference between the *stack* ADT and a *queue* ADT?

   Stack: last in first out

   Queue: first in first out

2. **(2 marks)** Consider the following algorithm.



```
Algorithm MIN1(A[0..n-1])
        If n = 1 then return A[0]
        else temp ← MIN1(A[0..n-2])
            if temp <= A[n-1] then return temp
            else return A[n-1]
            end if
        end if
```

   i. What does this algorithm compute?

      Return the minimum number in A[].

   ii. What is its basic operation?

      The first line “if n=1” is the basic operation because this line runs the most times.

   iii. How many times is the basic operation executed?

      n

   iv. What is the efficiency class of this algorithm? Show your working.

      C(n) = C(n-1)+1

      C(n-1) = C(n-2)+1

      ……

      C(1) = 1

      Thus C(n) = n and the complexity is O(n)

3. **(2 marks)** Use the Master Theorem to find the order of growth for the recurrence

$$T(n) = 4T(n/2) + n^2$$

According to Master Theory:

a = 4, b = 2, d = 2

a = b^d

So the complexity will be **(n^2)*logn**

4. **(1mark)** What is the worst case complexity of selection sort? Use big-Oh notation.

O(n^2)

5. **(1 mark)** What is the best case complexity of merge sort? Use big-Oh notation.

O(nlogn)

6. **(4 marks)** For each of the following statements, indicate whether it is true or false:

a.	"Heap sort is a stable sorting algorithm".	False
b.	"The worst case complexity of merge sort is $O(n^2)$.	False
c.	"Insertion sort is a stable algorithm".	True
d.	"The height of a complete binary tree with N nodes is N".	False

7. **(2 marks)** Explain briefly when a sorting algorithm is said to be *in-place*.

If it does not require additional memory except, perhaps, for a few units of memory

(from Lecture 5 page 5)

Section B (32 marks): Answer all the questions

8. (**3 mark)** Sequential search can be used with about the same efficiency whether a list is implemented as an array or a linked list. Is this statement true or false for binary search? Justify your answer.

It's false. For binary search in array, the complexity is O(logn), but for that in linked list, the complexity is O(nlogn) in the worst case.

9. **(3 marks)** You are managing a software project that involves building a computer-assisted instrument for medical surgery. The exact placement of the surgical knife is dependent on a number of different parameters, usually at least 25, sometimes more. Your programmer has developed two algorithms for positioning the cutting tool, and is seeking your advice about which algorithm to use:

Algorithm A has an average – case run time of *n*, and a worst case run time of n^4, where *n* is the number of input parameters.
Algorithm B has an average case run time of $n(\log n)^3$, and a worst case of n^2.

Which algorithm would you favor for inclusion in the software? Justify your answer.

I prefer algorithm B.
(The reason…….was deleted because I don't want to show my poor English here.)

10. **(4 marks)** This question is about graph algorithms

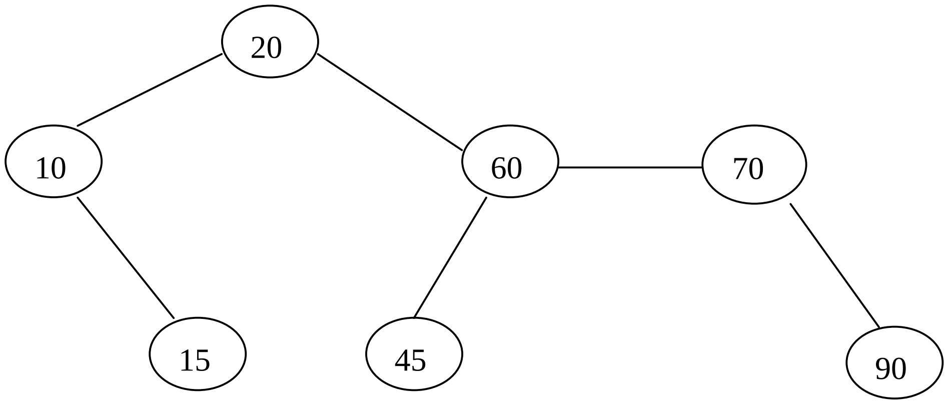
a. **(2 marks)** Sketch a graph that could be used to model the friendship network of a group of four people. Draw the corresponding adjacency matrix representation of your graph.

(a,b) is 1 representing a and b are friends. They are not friends if (a,b) is 0.

	a	b	c	d
a	-	1	1	1
b	1	-	1	1
c	1	1	-	1
d	1	1	1	-

Fine…all of them are friends for each other. Good story.

b. **(2 marks)** Traverse the graph by Breadth First Search (BFS). You may start the traversal at vertex 10 and construct the corresponding BFS forest:



```
    10
   /   \
15      20
          \
           60
          /   \
        45      70
                  \
                   90
```

The order is 10,15,20,60,45,70,90

11. **(5 marks)** This question is about heaps and heap sort.

   a. **(3 marks)** Sort the following list (into increasing order) by heap sort, using array representation of heaps. Show your workings. 13, 21, 16, 40, 30, 2, 10

   [13,21,16,40,30,2,10]

   construct heap -> [40 30 16 21 13 2 10]

   swap the max one and the last one in heap->[10 30 16 21 13 2 | 40]

   rebuild -> [30 21 16 10 13 2 | 40]

   swap -> [2 21 16 10 13 | 30 40]

   rebuild -> [21 13 16 10 2 | 30 40]

   swap -> [2 13 16 10 | 21 30 40]

   rebuild -> [16 13 2 10 | 21 30 40]

   swap -> [10 13 2 | 16 21 30 40]

   rebuild -> [13 10 2 | 16 21 30 40]

   swap -> [2 10 | 13 16 21 30 40]

   rebuild -> [10 2 | 13 16 21 30 40]

   swap -> [2 | 10 13 16 21 30 40]

   finish -> [2 10 13 16 21 30 40]

   b. **(2 marks)** State whether the following statements are true or false:

      i. A heap is a complete binary tree. True

      ii. The heap structure must satisfy either the structural constraint or the value relationship constraint. True

12. **(8 marks)** This question is about search trees

   a. **(2 marks)** What is the relationship between the value stored at a node and the values stored at its children in a Binary Search tree?

   The value stored at a node is bigger than the value in its left child and smaller than the value in its right child.

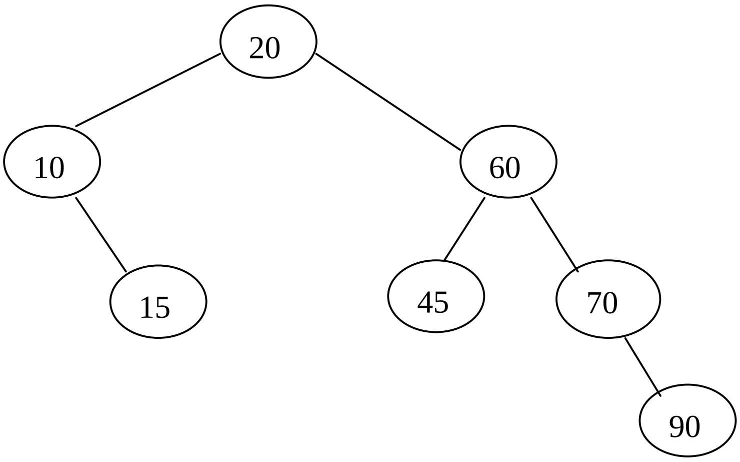
   b. **(2 marks)** Suggest one limitation of using a Binary Search Tree in search applications?

   For example, if we want to search all the values who can be divided by 7, then the binary search tree is useless.

   c. **(2 marks)** Build an AVL tree for the list of items: 1 6 2 1 5 8 7 4 2

```
     6
    /  \
   2    8
  /  \    \
 1    5    7
     /
    4
```

d. **(2 marks)** Show one possible binary search tree that can result from deleting 60 in the following binary search tree:



```
        20
      /    \
    10       70
      \    /    \
      15  45     90
```

13. **(9 marks)** This question is about hashing.

a. Insert items with the following keys into a hash table of size 7

12 7 6 8

Use the hash function h(k) = k(k+3) mod 7.

i. **(2 marks)** Use separate chaining for collision resolution. Show the hash value you have calculated for each input key, and show the hash table after all items have been inserted.

Hash()	0	1	2	3	4	5	6
Value1	7				8	12	
Value2						6	

ii. **(4 marks)** Use linear probing for collision resolution. Show the hash value you have calculated for each input key, and show the hash table after all items have been inserted.

Hash()	0	1	2	3	4	5	6
Value	7				8	12	6

iii. **(1 mark)** For the resulting hash table using linear probing, how many probes are needed in a search for the key 5.

4

b. **(2 marks)** Why is it not a good idea for a hash function to depend on just one letter (say the first one) of a natural language word?

Because words in natural language are not even about the first letter. For example, there are many words start with ‘T’, but only few words start with ‘X’.

Section C (25 marks) Answer all the questions

14. (6 marks) Write a correct and detailed algorithm that takes as input a Binary Search Tree (BST) T and finds the second largest value in the given BST. You may assume that the BST has at least two nodes, and that all of the node values are distinct. You must use clear, appropriately commented pseudo-code, Java or C to write your algorithm. Then, analyze the time and space complexity of your algorithm. Show your working for the complexity analysis.

```
Function find(T)
    Node max = T.root
    While max.right != null
        previous = max
        max = max.right
    if max.left != null
        return max.left.value
    return previous.value
```

Time: O(h), where h is the height of the tree, because in every loop, the algorithm reach a deeper level of the tree.

Space: O(1). There are only two new variables: max and previous.

15. (9 marks) Building the east-west road link created a budget blowout for the Victorian Government even before the project started. However, in the quest to provide better road infrastructure for suburbs in Melbourne, the Victorian Government in partisan with the road authorities, have now decided to lay bus routes connecting suburbs between the east and west of Melbourne. The road authorities have asked the project manager to start work on this project. The project manager has called on you – the software engineer – to design and implement an algorithm that manages a bus router problem between suburbs in Melbourne. The road authorities want to have bus routes in the suburb such that there is at least one bus route connecting every pair of suburbs. Furthermore, as laying roads is very expensive, the road authorities want to minimize the total amount of bus routes they wish to lay. Your algorithm must provide the project manager with a shorted bus route layout such that all suburbs between the eastern and western suburbs are connected, thus lowering costs.

a. What A*bstract Data Type (ADT)* should be used to help manage the bus-router problem?

   Graph.

b. Write a clear, correct and detailed algorithm to solve this problem. Make sure you design the algorithm to accept relevant input and; produces an appropriate output that the project manager can use to identify the shortest bus route layout connecting suburbs. You must use clear, appropriately commented pseudo-code, Java or C to write your algorithm.

In fact I'm not sure whether I understand this question correctly. If I understand it correctly, just use the prime algorithm in the lecture 20?

16. (10 marks) Let *A[0 ..n-1]* be an array of *n* distinct numbers. If *i < j* and *A[i] > A[j]*, then the pair *(i, j)* is called an ***inversion*** of A.

a) List the inversions of the array (2, 3, 8, 6, 1).

(0,4) (1,4) (2,4) (3,4) (2,3)

b) What arrays of size *n* have the largest number of inversions and what is this number?

The sorted array which is in decreasing order have the largest number of inversions. The number is (n-1)+(n-2)+…+1 = n*(n-1)/2

c) What arrays of size *n* have the smallest number of inversions and what is this number?

The sorted array which is in increasing order have the smallest number of inversions. The number is 0.

d) Write two different algorithms that you could use to count the number of inversions in a list of *n* elements, one based on each of the following design strategies:

(i) Brute force

(ii) Divide and conquer

You must use clear, appropriately commented pseudo-code, Java or C to write your algorithm.

Analyze the complexity of each algorithm and determine the efficiency class. You must show how you arrived at your answer.

(i)
```
Function BruteForce(A[])
    answer = 0
    For i = 0 to A.length-2
        For j = i+1 to A.length-1
            If A[i]>A[j]
                answer = answer+1
    return answer
```

Complexity: O(n^2) because there are two loop. The first loop will run n-1 times and the second loop will run (n-2)+(n-3)+ … +1 = (n-1)*(n-2)/2 which belongs to O(n^2).

(ii) Same with question 48 in week 7 tutorial exercises. Tutor's answer is on LMS