The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, FULL YEAR, 2019–2020

ALGORITHMS CORRECTNESS AND EFFICIENCY

Time allowed TWO hours

Candidates may complete the front covers of their answer books and sign their desk cards but must NOT write anything else until the start of the examination period is announced.

Answer all FOUR questions. The total mark is 100.

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject-specific translation directories are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn examination paper over until instructed to do so

Question 1 This question is about algorithm correctness. (20 marks)

(a) Explain the meaning of 'partial correctness' and 'total correctness' of algorithms. Give an example to illustrate the difference between 'partial correctness' and 'total correctness'. (4 marks)

(b) The following program implements a sorting algorithm:

```
public static void sort(int[] arr) {
     int len = arr.length;
     for(int i = len - 1; i > 0; i--){
          for(int j = 0; j < i; j++){
               if(arr[j] > arr[j+1]){
                   int temp = arr[j];
                   arr[j] = arr[j+1];
                   arr[j+1] = temp;
               }//end if
          } // end inner for loop
     }//end outer for loop
}
```

(i) The partial correctness of the program above can be proved by proving the loop invariant of the inner for loop and the loop invariant of the outer for loop. State a loop invariant for the inner for loop that is useful in proving the program correct. State a loop invariant for the outer for loop that is useful in proving the program correct. You may write your answer using either logical expressions or as pseudo-code. (6 marks)

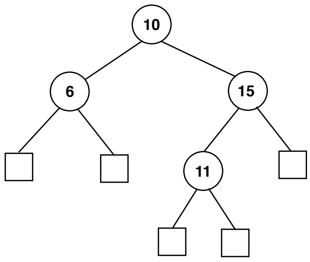
(ii) Prove the loop invariant of the inner for loop by mathematical induction. (8 marks)

(iii) Does the program above terminate? Justify your answer briefly. (2 marks)

Question 2 This question is about search tree structures. (30 marks)

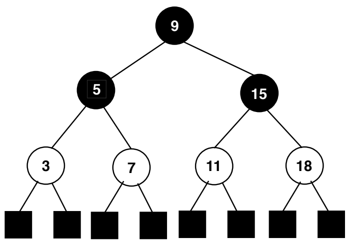
(a) What is a binary search tree? What is an AVL tree? (3 marks)

(b) Explain and draw figures to show the process of inserting the key 12 into the following AVL tree. Make sure that the resulting tree is still an AVL tree. (7 marks)

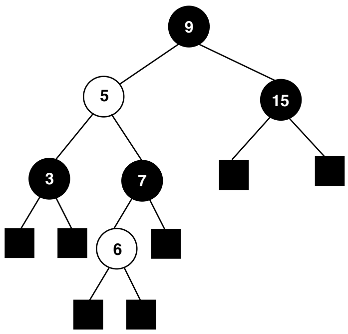


(c) What is a red-black tree? Describe the four properties that a red-black tree should satisfy. (4 marks)

(d) Explain and draw figures to show the process of inserting the key 12 into the following red-black tree. Make sure that the resulting tree is still a red-black tree. (6 marks)



(e) Explain and draw figures to show the process of deleting the key 15 from the following red-black tree. Make sure that the resulting tree is still a red-black tree. (10 marks)



Question 3 This question is about sorting algorithms, heaps and hash tables. (25 marks)

(a) In in-place quick sort, two indices j and k are used to partition the sequence S into L and $E \cup G$, where L, E, G are the sets of elements that are less than, equal and greater than the pivot, respectively. Use in-place quick-sort partition to partition the follow sequence:

34 j	62	96	45	17	80	50 k	60

The initial positions of these two indices are shown above, and the pivot is 60. You should show the intermediate steps and the final partition results. (5 marks)

(b) Let PQ be a sorted priority queue. It is initially empty. State the contents of the PQ after each of the following operations: (8 marks)

Method	Priority Queue Contents
insert(4,B)	
insert(8,A)	
insert(2,C)	
min()	
insert(8,D)	
removeMin()	
removeMin()	
removeMin()	

(c) PQ-sort scheme can be used to implement several sorting algorithms. Briefly describe how the priority queue could be used to implement selection sort, insertion sort and heap sort. State the time complexity of these three sorting algorithms in terms of Big-O notation. (6 marks)

(d) For double hashing, given an initially empty hash table of size 13, hash functions $h(k) = k \mod 13$, $d(k) = 7 - (k \mod 7)$, insert the following keys into the hash table. (You may show the final hash table only.) 15, 42, 54, 29, 91, 30. (6 marks)

Question 4 This question is about graphs, string matching and dynamic programming. (25 marks)

(a) Consider the directed graph given by the following adjacency lists:

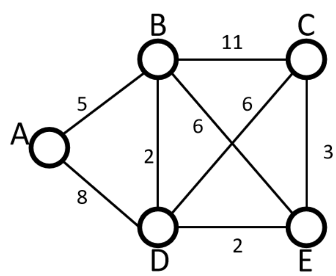
```
A ---> {B, D}
B ---> {C}
C ---> {E, F}
D ---> {B, E}
E ---> {B, F}
F ---> {}
```

(i) Is this graph a connected graph? Is it a strongly connected graph? (2 marks)

(ii) Trace a **breadth-first** traversal of the graph starting from A using a queue. Show which nodes are in the queue at each step. Finally, list the nodes in the order that they are marked as "visited". Note that if a node has more than one unvisited neighbour, choose them in the alphabetical order. (7 marks)

(iii) Start from A, use depth-first-search to detect cycles in the graph. You should show which nodes are in the stack for each step. (3 marks)

(b) Given the weighted graph shown below,



(i) Find the sum of distances of the shortest paths from node A to all nodes. (3 marks)

(ii) Find the sum of the weights of all edges of a minimum spanning tree. (3 marks)

(c) This question is about pattern matching algorithms.

(i) Given the charcater set as S = {a, f, g, i, l, m, n, o, t}, find the last-occurrence function for pattern "amalgamation" for the Boyer-Moore algorithm. (2 marks)

c	a	f	g	i	l	m	n	o	t
L(c)									

(ii) Given the pattern "amalgamation", find the fail function for this pattern for the KMP algorithm. (2 marks)

c	a	m	a	l	g	a	m	a	t	i	o	n
f(c)												

(d) Given matrix chain-product $A_1 * A_2 * A_3 * A_4$ and the size of matrices shown below, find its optimal parenthesization, and the minimum number of scalar multiplications. (3 marks)

Matrix	Size
A_1	20×10
A_2	10×5
A_3	5×2
A_4	2×40