The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, FULL YEAR, 2018–2019

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. (25 marks)

(a) Recall that the Assignment rule in the proof calculus is:

$$\frac{}{\{\psi[E/x]\}\ x = E\ \{\psi\}}\ \text{Assignment}$$

Given the program statement:

$$x = x - 1$$

use the Assignment rule to derive the precondition corresponding to each of the following postconditions (it is not necessary to simplify the precondition):

(i) $x = 2$

(ii) $x = y + 1$

(iii) $x * x = x + 1$

(iv) $(x < y) \wedge (x > z)$

(4 marks)

(b) What is meant by the terms invariant and variant in relation to the correctness of algorithms? Explain the meaning of partial and total correctness of algorithms briefly and the role of invariants and variants in establishing partial and total correctness of algorithms. (4 marks)

(c) 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--){
          int pos_greatest = 0;
          for(int j = 0; j <= i; j++){
               if( arr[j] > arr[pos_greatest]){
                 pos_greatest =  j;
               }//end if
           }//end inner for loop
          int temp = arr[i];
          arr[i] = arr[pos_greatest];
          arr[pos_greatest] = temp;
     }//end outer for loop
}
```

(i) What is the name of this algorithm? Trace it for the following array: $[3, 1, 5, 2]$. (4 marks)

(ii) State an invariant for the inner loop that is useful in proving the algorithm correctness. State an invariant for the outer loop that is useful in proving the algorithm correctness. You may write your answer either in first-order predicate calculus (using quantifiers) or English. (5 marks)

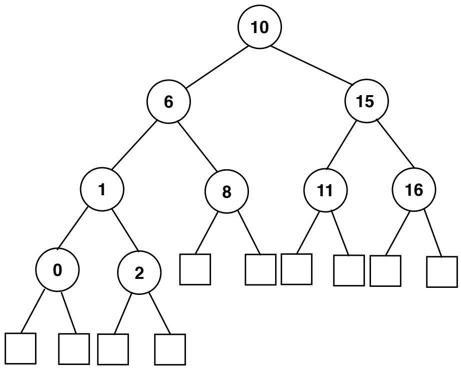
(iii) Prove the loop invariant of the inner for loop by mathematical induction. (5 marks)

(iv) Does the program above terminate? Justify your answer briefly. (3 marks)

Question 2 This question is about search tree structures. (25 marks)

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

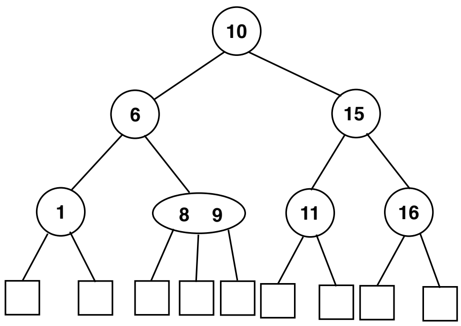
(b) Explain and draw figures to show the process of inserting the key 5 into the following AVL tree. Make sure that the resulting tree is still an AVL tree. (7 marks)



(c) Let n denote the number of entries in an AVL tree. Analyze the time complexity of inserting an entry into an AVL tree using the big-Oh notation. The time complexity of the main steps involved in the insertion process should be presented in the answer. (6 marks)

(d) What is a $(2,4)$ tree? (2 marks)

(e) Explain and draw figures to show the process of deleting the key 11 from the following $(2,4)$ tree. Make sure that the resulting tree is still a $(2,4)$ tree. (6 marks)



(f) Let n denote the number of entries in a $(2,4)$ tree. What is the time complexity of applying a deletion operation in a $(2,4)$ tree? Present it using the big-Oh notation. (1 mark)

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

(a) Let Q be a queue. It is initially empty. A total of 32 *enqueue* operations and 15 *dequeue* operations are performed on Q. 5 of these *dequeue* operations return null to indicate an empty queue. Find the current size of Q.

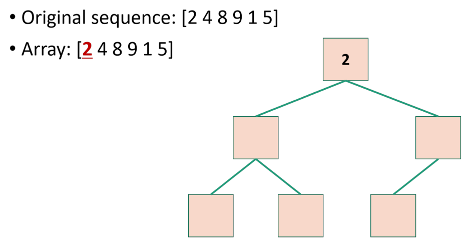
(2 marks)

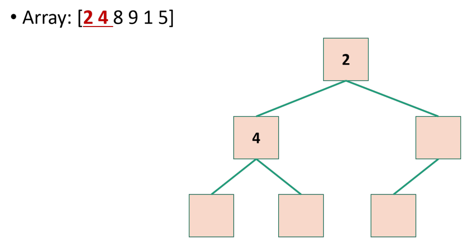
(b) Design an efficient divide-and-conquer algorithm to find the median in $O(n)$, and explain why the time complexity of your algorithm is $O(n)$. Illustrate your algorithm using the following example.
[3, 5, 2, 1, 2, 7, 8, 9, 0, 3, 9, 8, 4, 1, 6]

(7 marks)

(c) Use the in-place heap sort algorithm to sort array [2, 4, 8, 9, 1, 5] in an increasing order. The first two steps are given below. You should strictly follow these steps to sort the array. You should show the content of the array and the heap in all intermediate steps. Note that you should modify the heap to maximum-oriented heap for in-place heap sort.

(9 marks)





(d) Insert the following keys into a hash table of size 13 using linear probing and double hashing, respectively. Initially the hash table is empty.
15, 42, 18, 54, 91, 77, 51, 30, 64.

(i) For linear probing, use the following hash function $h(x) = x \bmod 13$. Show the resulting hash table.

(2 marks)

(ii) For double hashing, use the following hash functions: $h(k) = k \bmod 13$, $d(k) = 7 - (k \mod 7)$, i.e. place a key in the first available cell of the series $(h(k)+jd(k)) \mod N$, for $j = 0, 1, \ldots, N-1$. Fill in the tables below.

k	$h(k)$	$d(k)$	Probes		
15					
42					
18					
54					
91					
77					
51					
30					
64					
			j=0	j=1	j=2

0	1	2	3	4	5	6	7	8	9	10	11	12

(3 marks)

(iii) Which method, linear probe or double hashing, is more preferable to handle collision? You should well justify your answer.

(2 marks)

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

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

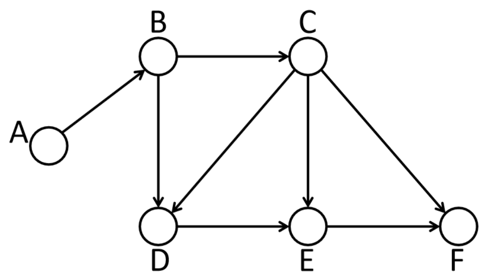
```
A ---> {B, D}
B ---> {C}
C ---> {E, F}
D ---> {B, E}
E ---> {B, F}
F ---> {}
```

(i) Write down the adjacency matrix for the graph above. (2 marks)

(ii) Trace a depth-first traversal of the graph starting from A using a stack. Show which nodes are in the stack 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 one in the alphabet order. (5 marks)

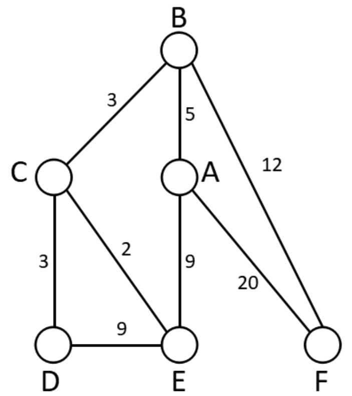
(iii) Start from A, use depth-first-search to detect cycles in the graph. You should show which nodes are in the stack for each step. (3 marks)

(b) Write down the topological order of the following graph. (2 marks)



(c) Given the weighted graph shown below,

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



(ii) Find the sum of the weights of all the edges in the minimum spanning tree.

(3 marks)

(d) This question is about pattern matching algorithms.

(i) Given the charcater set as S = {a, b, c, d, e, f, g, r, s, t}, find the last-occurrence function for pattern "abstract" for Boyer-Moore algorithm.

c	a	b	c	d	e	f	g	r	s	t
L(c)										

(2 marks)

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

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

(2 marks)

(e) Given the matrix chain-product as $A_1 * A_2 * A_3 * A_4$, and the size of matrices shown below, find the optimal parenthesization for this matrix chain-product with the minimum number of scale multiplications.

Matrix	Size
A_1	30×40
A_2	40×5
A_3	5×10
A_4	10×20

(3 marks)