The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, FULL YEAR 2020–2021

Algorithms Correctness and Efficiency

Time allowed Two hours

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

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. (24 marks)

```
1   public static void sort(int[] arr) {
2        int len = arr.length;
3        for(int i = 0; i < len-1; i++){
4             int pos = i;
5             for(int j = i; j <= len-1; j++){
6                  if( arr[j] < arr[pos]){
7                      pos = j;
8                  }//end if
9             } // end inner for loop
10            int temp = arr[pos];
11            arr[pos] = arr[i];
12            arr[i] = temp;
13       }//end outer for loop
14  }
```

(a) Trace the program above for the array $arr = [3, 5, 2, 1]$. Describe how the value of *pos* and the content of the array change. (3 marks)

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

(c) 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)

(d) 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. What is the loop invariant of the inner for loop? What is the loop invariant of the outer for loop? You may write your answer using either logical expressions or their equivalent English expressions. (6 marks)

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

Question 2 This question is about search tree structures. (26 marks)

(a) What is a (2, 4) tree? What is an AVL tree? (4 marks)

(b) Show and explain the steps with appropriate figures in deleting the entry with key 5 from the $(2, 4)$ tree in Figure 1. (7 marks)

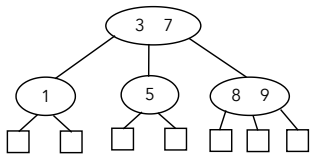

Figure 1: A (2, 4) tree

(c) Show and explain the steps with appropriate figures in deleting the entry with key 2 from the AVL-tree in Figure 2. (7 marks)

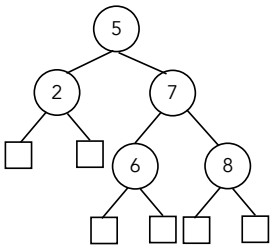

Figure 2: An AVL tree

(d) Suppose we implement the Sorted Map ADT using an AVL tree as its underlying data structure. Describe the main steps of $ceilingEntry(Key\ k)$ using pseudocode. What is the time complexity of $ceilingEntry(Key\ k)$? Denote it using Big-Oh notation. (8 marks)

```
Entry ceilingEntry(Key k): returns the entry with the
least key value greater than or equal to the given key k;
returns null if no such entry exists.
```

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

(a) Divide-and-conquer is a general algorithm-design paradigm, consisting of three steps: divide, recur and conquer. Describe how the quick-sort algorithm follows the divide-and-conquer paradigm regarding to these three steps. (6 marks)

(b) Let PQ be an initially empty priority queue. A total of 30 *insert(k,v)*, 10 *min()* and 15 *removeMin()* operations are performed on PQ. 5 of these *removeMin()* operations return *null* to indicate an empty priority queue. Find the current size of PQ. (3 marks)

(c) A heap is a binary tree storing keys at its nodes, which satisfies two heap properties. State these two heap properties and briefly describe them. (4 marks)

(d) 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) Is it a strongly connected graph? (2 marks)

(ii) Start from A, use depth-first-search to detect cycles in the graph. Show which nodes are in the stack for each step. (4 marks)

(e) Given the weighted graph shown in Figure 3 below,

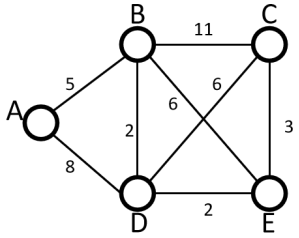

Figure 3: A weighted graph.

(i) Find the shortest distances from node A to all nodes, and their summation. (3 marks)

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

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

(a) This question is about pattern matching algorithms.

(i) Given the character set S = {a, e, f, i, l, m, n, o, p, r, s, t, x}, find the last-occurrence function for "xenotransplantation" of the Boyer-Moore algorithm. (2 marks)

c	a	e	f	i	l	m	n	o	p	r	s	t	x
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)												

(b) Assume the total number of strings in $\boldsymbol{S}$ is n, the number of characters of the string to be searched is m and the size of alphabet is d, what is the time complexity in (Big-Oh notation) of a standard trie to search the string in $\boldsymbol{S}$? (2 marks)

(c) Dynamic programming often applies to a problem that at first seems to require a lot of time. State three essential criteria of the problem where dynamic programming techniques could be applied, and briefly explain each criterion using one sentence. (6 marks)

(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. (4 marks)

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

(e) Given a 10-step staircase, $s_0, s_1, s_2, \ldots, s_{10}$, a person can only step one or two steps at a time, e.g. from s_0 to s_1 or from s_0 to s_2, but not from s_0 to s_3. The objective is to find the number of distinct paths for a person to go from s_0 to s_{10}.

(i) Denote each step of the staircase as a vertex, and possible move from one step to another as an edge, e.g. if a person can move

from s_0 to s_1, there will be a directed edge from s_0 to s_1. Model the problem above as an unweighted directed graph and draw the graph, assume starting from s_0 to s_{10}. (2 marks)

(ii) Denote the number of distinct ways from Step x to Step 10 as $n[x]$, write down the characterizing equation when solving it using dynamic programming. (3 marks)

(iii) Find the number of distinct paths for a person to go from Step 0 to Step 10 by filling in the table below. (The values of n[10], n[9] and n[8] are given as an example.) (4 marks)

x	0	1	2	3	4	5	6	7	8	9	10
$n[x]$									2	1	0