The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, FULL YEAR, 2023–2024

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 mathematical methods for analysing algorithm correctness and efficiency. [25 marks]

(a) Let $f(n)$ and $g(n)$ be functions mapping positive integers to positive real numbers. Explain the meaning of '$f(n)$ is $\Theta(g(n))$'. (2 marks)

(b) Consider the following program:

```
void f(int m){
     ...
}
void g(int m){
     ...
}
void h(int n){
     int p = 1;
     for (int i = 1; i <= n; i++){
         f(i);
         if(i == p){
              g(n);
              p *= 2;
         }
     }
}
```

If the time complexity of $f(m)$ and the time complexity of $g(m)$ are both $\Theta(m)$, then what is the time complexity of $h(n)$ as a function of n? Denote the time complexity of $h(n)$ using Big-Oh and justify your answer. (4 marks)

(c) Explain the meaning of 'partial correctness' and 'total correctness' of algorithms. (4 marks)

(d) Let $Find(A)$ be an algorithm which takes an array A of length n as input and returns an index i. Any index in A is an integer greater than or equal to 0 and less than n.

(i) Specify the postcondition of $Find(A)$ using predicate logic to say that i is the index of a *maximal* element in A.

(ii) Specify the postcondition of $Find(A)$ using predicate logic to say that i is the index of the *first minimal* element in A.

(5 marks)

(e) Given the program statement: $x = y + z$, use the 'Assignment rule' in the proof calculus (see Appendix A) to derive the precondition corresponding to each of the following postconditions (it is not necessary to simplify the precondition):

(i) $x = y + 1$

(ii) $x * n = x!$

(iii) $\forall x.(x > 0)$

(iv) $\exists y.(x < y + z)$

(4 marks)

(f) Let P be the program:

$$
\begin{array}{l}
y = 0; \\
\textit{while}\ (y\ \ != x)\{ \\
\qquad y = y + 1; \\
\}
\end{array}
$$

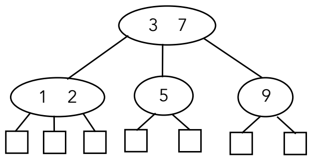
Prove that $\vdash_{par} \{x \geq 0\}\ P\ \{x = y\}$ is valid using the proof calculus.

(6 marks)

Question 2 This question is about search tree structures. [25 marks]

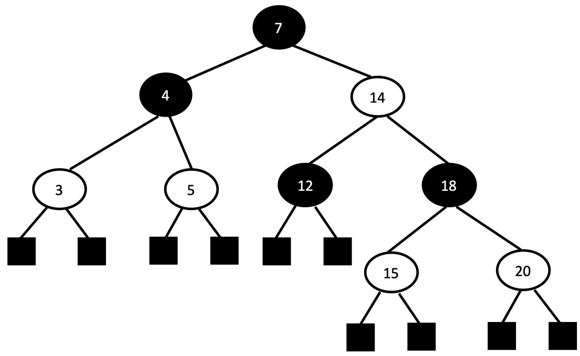
(a) What is a binary search tree? What is a (2, 4) tree? (4 marks)

(b) Show and explain the steps with appropriate figures in deleting the entry with key 5 from the $(2, 4)$ tree below. (5 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 17 into the following red-black tree. Make sure that the resulting tree is still a red-black tree. (6 marks)



(e) Suppose we implement the Sorted Map ADT using a red-black tree. Describe the main steps of *floorEntry*(*Key k*) using pseudocode. What is the time complexity of *floorEntry*(*Key k*)? Denote it using Big-Oh notation. (6 marks)

```
Entry floorEntry(Key k): returns the entry with the
greatest key value less 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 graphs. [26 marks]

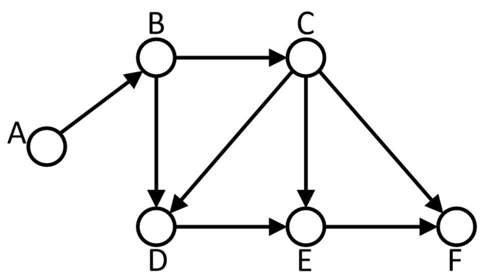
(a) Write down the pseudo-code of Quick-Sort, given the input sequence S. (8 marks)

(b) For double hashing, use the following 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, 18, 54, 91, 77, 51, 30, 64.
Please answer the question in the answer booklet instead of writing directly on the question sheet. (9 marks)

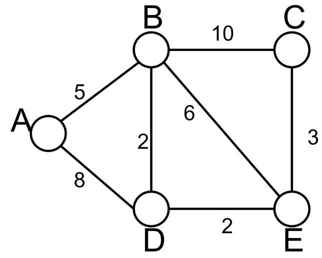
k	$h(k)$	$d(k)$	Probes		
15					
42					
18					
54					
91					
77					
51					
30					
64					
			j=0	j=1	j=2

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

(c) Write down the topological order of the following graph. (3 marks)



(d) Given the weighted graph shown below,



(i) Find the shortest distances from node A to all nodes, and their sum. (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. [24 marks]

(a) Given the following string:
What a lovely day!
How many character matches are needed to find the pattern "day!" from the string using the Boyer-Moore Algorithm? (3 marks)

(b) Construct the suffix trie for "banana", and briefly explain why you need to add "$" at the end of the string. (5 marks)

(c) 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. (6 marks)

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

(d) There is an array of integers $[a_0, a_1, \ldots, a_{N-1}]$. You can select some of these integers, but they must not be adjacent. Find the maximum sum you can select by using dynamic programming (DP).

(i) Define the subproblem in your DP solution. (2 marks)

(ii) Define the recurrence function in your DP solution. (4 marks)

(iii) Given the following array, find the maximum sum of non-adjacent integers. Show your working.
5 3 1 3 4 1 2 1 (4 marks)

Appendix A: Proof Rules in the Proof Calculus

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

$$\frac{\{\phi\}C_1\{\eta\} \quad \{\eta\}C_2\{\psi\}}{\{\phi\}C_1;C_2\{\psi\}}\ \text{Composition}$$

$$\frac{\{\phi \wedge B\}C_1\{\psi\} \quad \{\phi \wedge \neg B\}C_2\{\psi\}}{\{\phi\}\ \text{if } B \text{ then } C_1 \text{ else } C_2\ \{\psi\}}\ \text{If-statement}$$

$$\frac{\{\phi_1\}C_1\{\psi\} \quad \{\phi_2\}C_2\{\psi\}}{\{(B \to \phi_1) \wedge (\neg B \to \phi_2)\}\ \text{if } B \text{ then } C_1 \text{ else } C_2\ \{\psi\}}\ \text{If-statement}$$

$$\frac{\vdash \phi' \to \phi \quad \{\phi\}C\{\psi\} \quad \vdash \psi \to \psi'}{\{\phi'\}C\{\psi'\}}\ \text{Implied}$$

$$\frac{\{\psi \wedge B\}\ C\ \{\psi\}}{\{\psi\}\ \text{while } B \text{ do } C\ \{\psi \wedge \neg B\}}\ \text{Partial-while}$$

$$\frac{\{\eta \wedge B \wedge 0 \leq E = E_0\}\ C\ \{\eta \wedge 0 \leq E < E_0\}}{\{\eta \wedge 0 \leq E\}\ \text{while } B \text{ do } C\ \{\eta \wedge \neg B\}}\ \text{Total-while}$$