The University of Nottingham Ningbo China

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, FULL YEAR, 2016–2017

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 Multiple choice questions. For some questions, there is exactly one correct answer. For other questions, there are one or more correct answers. For each question, if *all* correct answers are selected and no incorrect ones are selected, then two marks will be awarded; otherwise, no mark will be awarded. (20 marks)

(a) Which one of the following is a concrete data structure rather than an abstract data type?

(i) list
(ii) linked list
(iii) stack
(iv) queue

(b) Choose the class(es) to which the function $S(n) = \log n \times (2^{(2 \log n)} + 1)$ belongs to. The base of the log is 2.

(i) $\Omega(\log n)$
(ii) $\Theta(n \log n)$
(iii) $O(n^2)$
(iv) $\Theta(n^2 \log n)$
(v) $O(n^3)$

(c) Suppose that an intermixed sequence of (stack) push and pop operations are performed. The push operations push the integers 0 through 5 in order; the pop operations print out the returned value. Which of the following sequence(s) cannot occur in the returned result?

(i) 1 2 3 4 5 0
(ii) 1 4 5 3 0 2
(iii) 4 3 2 1 0 5
(iv) 4 5 3 2 0 1
(v) 4 3 2 1 0 5

(d) 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. What is the current size of Q?

(i) 32

(ii) 27

(iii) 22

(iv) 17

(v) None of the above

(e) Which one of the following data structures has the *fastest* insertion procedure?

(i) binary search tree

(ii) heap

(iii) ordered linked list

(iv) unordered linked list

(f) Which one is the best estimate of the number of times that Line 5 is executed during a single call of $algorithm(n)$?

```
1   int algorithm (int n) {
2        int k = 0;
3        int m = 8n;
4        while (m>1){
5             m = m/2;
6             k++;
7        }
8       return k;
9  }
```

(i) 1

(ii) $3 \log n$

(iii) $3 + \log n$

(iv) $(\log n)^3$

(v) $8n$

(g) Suppose you are running an algorithm on a set of files. The running time of your algorithm is $O(n^3)$, where n is the number of files. For 300 files it takes 1 hour to complete. How long would you expect it to run on 600 files? Pick just one answer.

(i) 2 hours

(ii) 4 hours

(iii) 6 hours

(iv) 8 hours

(v) 60^3 minutes (which is 3600 hours)

(h) Which one of the following code fragments has $result = n^i$ as a loop invariant?

(i)
```
int power(int n, int k){
        int result = 1;
        for (int i = 1; i < k; i++){
              result = result * n;
        }
}
```

(ii)
```
int power(int n, int k){
        int result = n;
        for (int i = 1; i < k; i++){
              result = result * n;
        }
}
```

(iii)
```
int power(int n, int k){
        int result = n;
        for (int i = 0; i < k; i++){
              result = result * n;
        }
}
```

(iv)
```
int power(int n, int k){
        int result = n;
        for (int i = 0; i < k; i++){
              result = result * result;
        }
}
```

(i) Which one of the following data structures has reliably efficient (logarithmic) performance for search, insertion and deletion?

(i) ordered array
(ii) ordered list
(iii) unordered list
(iv) binary search tree
(v) balanced binary search tree

(j) The space complexity of breadth-first traversal of a graph using a queue is best represented by which one of the following:

(i) $O(1)$
(ii) $O(|E|)$ where $|E|$ is the number of edges
(iii) $O(|V|)$ where $|V|$ is the number of vertices
(iv) $O(|V| + |E|)$
(v) $O(2^{|V|})$

Question 2 This question is about sorting algorithms. (24 marks)

(a) 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: $[5, 3, 1, 2]$. (4 marks)

(ii) 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 as pseudo-code. (4 marks)

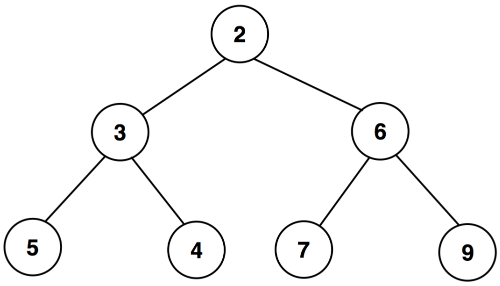
(iii) Prove the loop invariant of the inner for loop by mathematical induction. (6 marks)

(b) Divide-and-conquer is a general algorithm design paradigm, consisting of three steps: divide, recur and conquer. Describe how the merge-sort algorithm and the quick-sort algorithm follow the divide-and-conquer paradigm differently regarding to these three steps. (4 marks)

(c) What is the worst-case time complexity of quick-sort? When does the worst case for quick-sort occur?
What is the best-case time complexity of quick-sort? When does the best case for quick-sort occur?
What is the average-case time complexity of quick-sort? Give a brief justification of your answer. A full proof is *not* required. (6 marks)

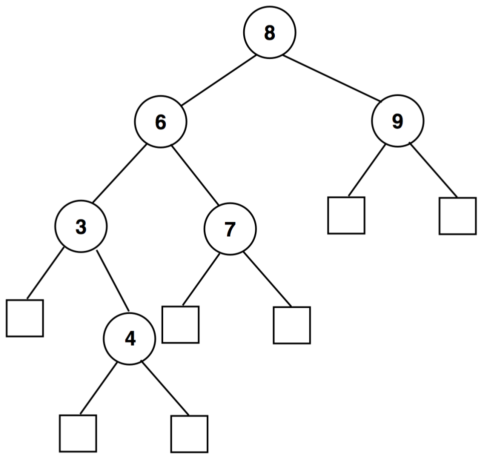
Question 3 This question is about priority queues and maps. (28 marks)

(a) What is a heap?
Explain and draw figures to show the process of inserting a key 1 into the following heap. (6 marks)



(b) Explain and draw figures to show the process of removing the minimal key from the heap drawn exactly as above. (4 marks)

(c) What is a binary search tree?
Explain and draw figures to show the process of inserting a key 5 into the following binary search tree. (6 marks)



(d) Explain and draw figures to show the process of removing the key 6 from the binary search tree drawn exactly as above. (4 marks)

(e) Suppose you have a hash table which consists of an array of size $N = 11$ and a hash function $h(k) = k \mod N$. Double hashing is used to handle possible collisions. The secondary hash function is $d(k) = 5-(k \mod 5)$. An item should be placed in the first available cell of the series $(h(k) + jd(k)) \mod N$, for $j = 0, 1, ..., N - 1$. Show the process of inserting the keys $35, 57, 39, 46$, in this order, into the hash table. The hash values for these keys are provided in the table below. The hash table is initially empty. (8 marks)

k	$h(k)$	$d(k)$
35	2	5
57	2	3
39	6	1
46	2	4

Question 4 This question is about graphs. (28 marks)
Consider the graph given by the following adjacency lists:

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

(a) Trace a breadth-first traversal of the graph starting from A using a queue. At each step, show which nodes are in the queue.
List the nodes in the order that they are marked as 'visited'.
(6 marks)

(b) Trace a depth-first traversal of the graph starting from A using a stack. At each step, show which nodes are in the stack.
List the nodes in the order that they are marked as 'visited'.
(6 marks)

(c) Trace a topological sort of the graph. At each step, display the state of the output array of nodes and the nodes and edges which remain in the graph. (4 marks)

(d) Explain how to modify the depth first search (DFS) to detect cycles in a graph. Write down the pseudo-code of the modified DFS.
(6 marks)

(e) Add one more edge to the graph, from E to B. Show the process of using the modified DFS in (d) to detect a cycle in the new graph.
(6 marks)