Name:_________________________________ Class:____________

JURONG PIONEER JUNIOR COLLEGE

JC2 Mid – Year Examination 2023

COMPUTING
Higher 2

9569/01

30 June 2023

Paper 1 (Written)

1 hour

Additional materials: Answer Paper
Cover Page

READ THESE INSTRUCTIONS FIRST

Answer papers will be provided with the question paper.
Write your name and civics class on all the work you hand in.
Write in dark blue or black pen on both sides of the paper.
You may use an HB pencil for any diagrams or graphs.
Do not use staples, paper clips, glue or correction fluid.

Answer **all** the questions.
Approved calculators are allowed.
You are reminded of the need for clear presentation in your answers.

At the end of the examination, fasten all your work securely together.
The number of marks is given in brackets **[ ]** at the end of each question or part question.
The total number of marks for this paper is **30**.

This document consists of **3** printed pages.

[Turn over

1 Study the following sorting code carefully.

```
def sort(lst):
    if len(lst)<=1:
        return lst
    else:
        pivot = lst[0]
        smaller = []
        larger = []
        for i in range (1, len(lst)):
            if lst[i] < pivot:
                smaller.append(lst[i])
            else:
                   larger.append(lst[i])

           return sort(smaller) + [pivot,] + sort(larger)
```

a) State the name of the above sorting algorithm. [1]

Quick Sort

b) Explain why the above sorting algorithm is inefficient when it is used on a nearly sorted array. [1]

The above algorithm will choose pivot from 1 side (the smallest) and partition the elements to smaller and larger groups. Since the elements are sorted, the distribution will be skewed to one (larger) side instead of dividing into 2 equal halves, causing the time complexity to increase. [1]

It will increase from $O(n \log_2 n)$ to $O(n^2)$

c) Explain how you can modify the code to improve the efficiency. [2]

Change pivot = lst[0] to pivot = lst[midptr] [1]

In this way, if the original data is sorted, the smaller and larger groups will be equally distributed, giving a time complexity of $O(n \log_2 n)$. [1]

Bubble sort and insertion sort are both used to sort a nearly sorted integer array of size 1000 and there are only 5 integers in the array that are not in the correct position.

d) State and explain why insertion sort generally perform better than bubble sort for nearly sorted array. [2]

Insertion sort performs fewer comparisons.

For bubble sort to correct one incorrect item in an array, for example a small value that is found at the back of the array, it will need to take many passes to bring the item to the correct position where each pass invokes n-1 comparisons. [1]

For insertion sort to correct one incorrect item in an array just like the example above, it only needs 1 pass. [1]

2 Consider the following hash function for a hash table that uses linear probing (i.e.: insert into the next available space) to resolve collisions:

```
FUNCTION Hash (key: INTEGER) RETURN INTEGER
     RETURN (key + 3) MOD 7
```

Assume that the hash table array has a capacity of 7 and initially all slots are empty.

a) Suppose the keys 5, 12, 1, 16, 17, 15 are inserted into the hash table in this order. Illustrate the resulting hash table array after all the insertions are completed. [1]

[0]	15
[1]	5
[2]	12
[3]	
[4]	1
[5]	16
[6]	17

b) If we now search for the key 3, what is the expected number of comparisons needed? Explain your answer. [3]

5 comparisons.
(3+3) MOD 7 = 6, start linear probing/ comparing from index 6.
Continue comparing until index 3 which is empty. Since 3 would have been stored in the next available space, 3 does not exist and comparison stops.

c) Explain why choosing a good hash function is important for the performance of a hash table. Give an example of a situation where a bad hash function could lead to poor performance. [2]

Good hash function minimal collision/ clustering. Hence, best case time complexity is O(1).
Bad hash function produces a high number of collisions, where keys are mapped to the same index in the hash table. This increases the search comparisons up to O(n).

3 The manager of a local restaurant engaged a consultant to propose a digital solution for his restaurant operations. After conducting a comprehensive study, the consultant proposed a web-based solution using a **client-server model**.

The solution requires the following hardware to access the web server **wirelessly** in the local area network (LAN):

1. A tablet device on each table for customers to browse the menu and order food items.

2. Multiple large monitors for the chefs in the kitchen to read the ordered food items.

3. A computer station for the service staff to check the table number before serving the food to the customers.

4. A computer in the manager’s office to update the menu in the web server and print the daily sales report.

When a customer decides to pay the bill, a QR code will be generated on the tablet device for him to scan and make online payment using his personal mobile device.

(a) Explain the meaning of the term client-server model. [1]
Client-server describe a communication between the client’s program requesting a service or resource from the server’s program.[1]

(b) Describe one advantage of client-server model compared to peer-to-peer. [1]
Centralization: Access rights and resources allocation is done by the server.

Proper management: All data is stored in at the same server, thus it is easier to find the files.

In a client–server model, network security and maintenance, such as data backups, can be managed centrally.   ANY one

The restaurant’s manager is also keen to expand his business to accept **online ordering** for takeaways. [5]

(c) Draw the network diagram for the proposed web-based solution and include all the required hardware for the restaurant to accept online ordering.

[2]

Diagram should include the following:
- switch [1] connecting the web server, manager's computer, computer station for service crew and kitchen large display [1] [3]
- printer to print receipts
- wifi access point [1] for customers' tablet device

To accept online orders: router [1] and modem [1] connecting to ISP for internet service

Data is transmitted using packet-switching within the LAN and Internet.

(d) Describe how data is transmitted to the server when dine-in customers make orders.
The order data is sent in packets from the tablet to the switch in the LAN which will forward them to the server[1] using MAC addresses[1] in the packet header.
Packets are sequenced and can be send along different routes[1].
Packets arrive not-in-order at destination are re-ordered.[1].

(e) Describe how data is transmitted when online customers make orders.
The order data is sent in packets from the online customers to the routers[1], which will forward them to other routers to the next hop on the route [1] until it reaches the server using the IP address[1] in the packet header.

(Any 3 points, including about packets ordering given in part (d))

In the client-server model, customer orders are put into a queue.

f) Explain how a queue works in this context [1]
When orders are requested by clients, the requests are put in a queue based on their time of request. Queue works on first-in-first-out basis and the client that has been served will be dequeued from the queue.

This queue can be implemented in object-oriented programming as a data structure. The implementation of this queue requires two integer variables and an array.

g) Name and describe the purpose of the 2 variables. [1]
The integer variables are head (front) and tail (rear), which is to track the position of the first and last element of the queue respectively

h) Name and briefly describe the implementation of the two methods of the queue data structure to insert and delete orders. [4]
Enqueue is to insert new orders at the tail (rear) of the array. [1]
Increment rear by 1, and insert new order at rear index of the array.[1]

Dequeue is to remove orders from the head (front) of the array. [1]
Increment head (front) by 1. [1]

END OF PAPER