NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

MIDTERM TEST FOR

Semester 1, AY2022/23

CS2040 – Data Structures and Algorithms

1 October 2022 Time allowed: 1.5 hours

STUDENT **NO. :**	A	0	1						

INSTRUCTIONS TO CANDIDATES

1. **Do NOT flip / turn over the test paper until you are told to do so**
2. Shade your **student number** in page 1 of the answer sheet. Do **NOT** write your name!
3. **COMPLETELY shade** the bubble for each answer using a fairly **dark pencil**, except for open-ended Q5, in which you may write legibly in either pen or pencil
4. **Do NOT rearrange, add/remove staples or add/remove pages** from the answer sheet. **Submit only the ENTIRE answer sheet** at the end of the test. It is your responsibility to ensure that you have submitted it, and submitted the correct answer sheet
5. If you fail to submit the correct answer sheet, fail to provide **correct particulars** or prevent the options from being **automatically detected** by software, we will consider it as if you did not submit your answers. In the best case, **marks will be deducted**
6. No extra time will be given at the end of the test for you to write your particulars, to shade or to fill in the answer sheet. You must do it **before** the end of the test
7. This paper consists of five (**5**) questions. Q1-4 are "MCQ", shade **at most one option per grid**. Write your code for Q5 in the allotted answer sheet. The question paper comprises ten (**10**) printed pages including this front page. The answer sheet comprises two (**2**) printed pages
8. This is an open-hardcopy-notes examination but **WITHOUT** electronic materials
9. Marks allocated to each question are indicated. Total marks for the paper is **45**
10. The use of electronic **calculator** is **NOT** allowed

Qn	***Max***	***Marks***
Q1ab	04	
Q2abcdefg	14	
Q3ab	10	
Q4	09	
Q5	08	
Total	**45**	

Q1 [4 marks == 2 x 2]

A Java class X has **only one**

We want to

`ArrayList<X>` **L** containing The implementation of class X cannot be modified

Q1a. We can write our own *in Java* **L** correctly

○ True	○ False

Q1b. We can utilize *Java* **L** correctly

○ True	○ False

Q2 [14 marks == 7 x 2]

Computer memory can be visualized as in the diagram on the right, with the top row from left to right being the first 4 spaces, the middle row being the next 4 spaces, … and so on. There are only 12 spaces in the example on the right

Suppose there are:

- D
- P

  as well as
- (blank) – unused space

Data can be stored in , or as a

Data *Example*

			P	P	P	P	P
P	P	P	P	P			
		P	P			P	
		P			P		

Data *Example*

							P
P	P					P	P
P				P	P	P	P
P					P	P	P
P							

In this illustration, the first

- The first
- The second
- The third and last

Notes:

- The sizes of memory, number of (P) (blank) illustrated here are just an **example**, and may not always be as such
- Every
- Algo using any space outside counts as taking up additional space
- You may safely ignore << O(N) space taken up by system's call-stack if you are using a recursive algo
- You may claim the average/expected time for quick sort if pivot selection is randomized

For each question in **Q2a-g,**choose the time complexity of the best algorithm needed to solve the problem:

○ O(log(log(N))	○ O(log(N))	○ $O(\sqrt{N})$	○ $O(\sqrt{N}\log(N))$	○ O(N)	○ O(N log(N))
○ $O(N^{1.5})$	○ $O(N^{1.5}\log(N))$	○ $O(N^2)$	○ $O(N^2\log(N))$	○ $O(2^N)$	○ O(N!)

Suppose our data is stored in a , and you are limited to using

Q2a. if the data is requires _________ time

Q2b. if there are **at least** requires _________ time

Q2c. if there are **exactly** requires _________ time

We still have our data store in a , but now

Q2d. If there are **exactly** requires _________ time

Q2e. If there are **exactly** requires _________ time

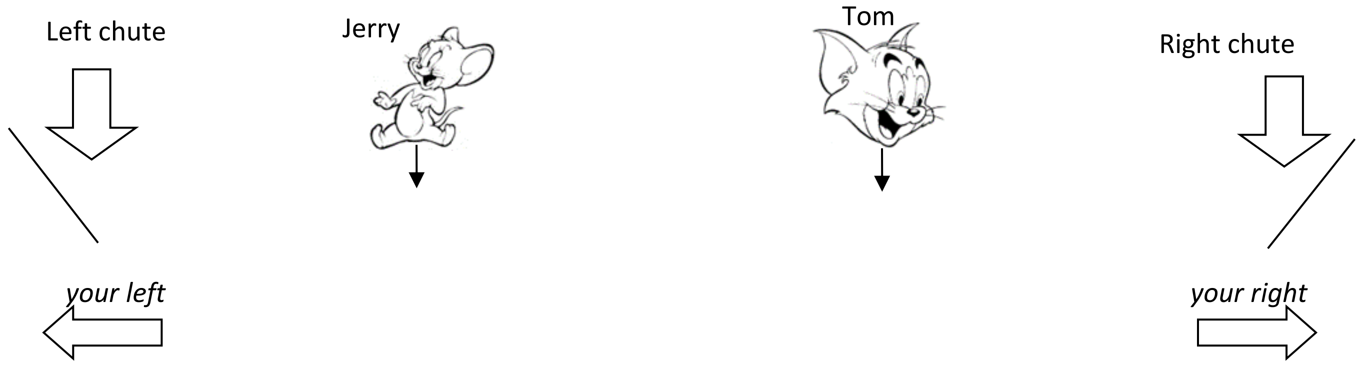
Q2f. If there are not more than requires _________ time

Q2g. If there are not more than requires _________ time

Q3 [10 marks == 2 x 5]

Jerry and Tom[1] are both working in a factory, standing along

This is illustrated in the diagram below:



The shown here are just an **example**

You want to implement a program that models these real-life operations:

- Item
  if
- **Jerry**
  if
- **Jerry**
  Output
  if
- **Jerry**
  if
- **Tom**
  output
  if
- **Output**
  if
- **Output**
  if

Each operation must be done in time. Deferring the cost to another operation is NOT acceptable here

You have the **choice** to keep track, or NOT keep track, of anything that you are not required to output

[1] Adapted from https://coloringonly.com/images/imgcolor/1548379186-tom-and-jerry-coloring-pages-lovely-tom-and-jerry-thumbs-up-coloring-page-tom-and-jerry-coloring-pages-of-tom-and-jerry-coloring-pages.jpg

For **each** of the question in **Q3a-b INDEPENDENTLY**, choose the minimal/simplest data structure(s) you need, i.e. **as far left an option as possible**, aside from O(1) spaced variables

Q3a. Only Cat/Dog/Cow(s) **WITHOUT** any extra

Cat –
Dog –
Cow –

○ 2 Cats	○ 1 Cat +1 Dog	○ 2 Dogs	○ 1Dog +1Cow	○ 2 Cows	○ 1 Dog + 2Cows

Q3b. Only Rat (R) and/or Wolf (W)

○ 2 R	○ 1 R + 1 W	○ 2 W	○ 2 R + 1 W	○ 1 R + 2W	○ 2 R + 2 W

BLANK SPACE

TILL END OF PAGE

Q4 [9 marks]

Match the following 4 code snippets to the 3 problems which are **independent** of one another. A problem will be correctly solved by 0 or 1 code snippet, while a code snippet will solve 0 or 1 problem

P1: Given the

P2: Given 2

P3: Given the

Each

For each of code snippets **Q4A-D** in **pseudocode**, shade the problem it **correctly solves** if any:

○ P1	○ P2	○ P3

Marks will be awarded for the entire Q4 instead of for individual parts. Shading option(s) when the code snippet does NOT solve those problem(s) *will be penalized*

BLANK SPACE

TILL END OF PAGE

Code Snippet **Q4A**

```
f(a ...                ) {





}
ans = f(a ...                );
```

Code Snippet **Q4B**

```
f(a ...                ) {





}
ans = f(a ...                );
```

Code Snippet **Q4C**

```
f(a ...                ) {





}
ans = f(a ...                );
```

Code Snippet **Q4D**

```
f(a ...                ) {





}
ans = f(a ...                );
```

Q5 [8 marks + 3?]

You have a diagram below
The class is defined below the diagram

ZZ is a Each object
has in the respective directions

```
class              {
      public
      public
}
```

Ivan is interested in It is
guaranteed that the will:

- NOT
- NOT
- NOT

If you perform this operation, which may be performed repeatedly, correctly in:

- O( ) time, you will get the full 8 marks and 3 bonus marks **(WARNING: time consuming)**
- O( ) time with O(1) space, you will get the full 8 marks
- O( ) time with O( ) space, you will get 5 marks
- O( ) time with O(1) space, you will get 3 marks

[If you have no time left, you might want to go for a lower-tier but correct solution to cut loss]

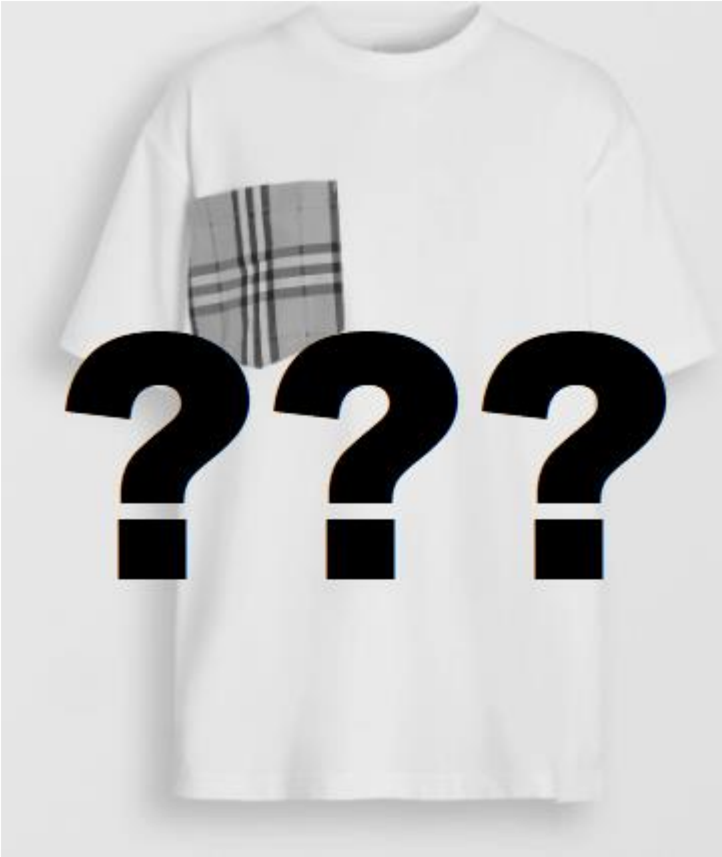
In the `mystery` method of the class, you are given

You are also given some other instance methods (that **MAY** or may **NOT** help you solve the problem?)

Your task is to **implement the `mystery` method** correctly and efficiently. You may implement other method(s) in the class if it helps you

```
class                   {


     public void           () {
          // your Q5 answer here
     }

     void                              {



     }

     void                              {




     }

     void                              {




     }

     void                              {




     }

     void                              {




     }
}
```

The earlier would look like the diagram below after the execution of `mystery( )` is completed



- End of paper -