NATIONAL UNIVERSITY OF SINGAPORE

CS2040 – DATA STRUCTURES AND ALGORITHMS

(Semester 1: AY2022/23)

Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. **Do NOT flip / turn over the test paper until you are told to do so**
2. Write your **student number** on pages 1, 3 AND 5 of the answer sheet. Do **NOT** write your name! Clearly **shade** your student number on page 1 too
3. Do NOT rearrange the answer sheet, or add/remove staples. **COMPLETELY shade** the bubble for each answer using a fairly **dark pencil**. You may answer the bonus question using pen or pencil
4. **Submit only the answer sheet** at the end of the assessment. 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 assessment for you to write your particulars, shade the answer sheet or write/transfer answers. You must do them **before** the end of the assessment
7. This paper consists of **17** questions. Not more than one option should be shaded per grid. The question paper comprises fourteen (**14**) printed pages including this front page and blank page 9. The answer sheet comprises five (**5**) printed pages, we will disregard the 6th page
8. This is an open-hardcopy-notes assessment but **WITHOUT** electronic materials
9. Marks allocated to each question are indicated. Total marks for the paper is **80**
10. The use of electronic **calculator** is **NOT** allowed

Sect Qn	***Max***	***Marks***
S1 Q1-7	32	
S2 Q8-12	15	
S3 Q13	8	
S3 Q14	8	
S3 Q15	10	
S3 Q16	6	
S3 Q17	1	
Total	**80**	

Section 1 – Shorts; Answer Questions 1-7 **[32 marks == 7 x 2 + 6 x 3]**

Question 1 – Warmup

A is a `java.util.` , **B** is a `java.util.` and **C** is a `java.util.` . Each of the 3 data structures contains **N** elements.

For **each** of **Q1a-g independent of one another**, the code snippet runs correctly, you are to choose the best answer for its time complexity:

◯ O(log(log(N))	◯ O(log(N))	◯ O($(\log(N))^2$)	◯ 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(N^3)

Code Snippet Q1a

```
System.out.println(                                    );
```

Code Snippet Q1b

```
...
```

Code Snippet Q1c

```
...
```

returns whose elements are

Code Snippet Q1d

```
...
```

Code Snippet Q1e

```
...                    // internally uses ...
...
```

Code Snippet Q1f

```
...
...
```

Code Snippet Q1g

```
...
...
```

`rand(0, A.size()-1)` runs in O(1) time and returns some integer from 0 to A.size()-1 inclusive.

Now, each of **Q2-7 is INDEPENDENT** of one another, and **worth more marks** than each part in Q1…

Question 2

In an , you are given 2 . The **maximum possible** *difference* in is:

◯ 0	◯ 1	◯ 2	◯ 3	◯ 4
◯ up to twice …			◯ to the moon!	

Question 3

You are given a in a . A is defined as:

```
class          {
   ...
   ...
}
```

You want to find the time or better, . Ali, Balu and Charlie each have a suggestion:

Ali – Start from

Balu – Since we have a

Charlie – Since we have a

Whose approach(es) work, if any at all?

◯ None	◯ Ali only	◯ Balu only	◯ Charlie only
◯ Ali & Balu only	◯ Ali & Charlie only	◯ Balu & Charlie only	◯ Ali, Balu and Charlie

Question 4

You have **N** *boxes* labelled 0..(**N**-1), each

You want to keep track of efficiently.

Ivan says "is suitable here…

X – a pair of

Y – given a

Z – BUT for fairly efficient operations, otherwise (time)."

You don't quite agree with Ivan though… Or do you?

◯ Disagree with all 3	◯ Agree with X only	◯ Agree with Y only	◯ Agree with Z only
◯ Agree with X, Y only	◯ Agree with X, Z only	◯ Agree with Y, Z only	◯ Agree with X,Y,Z

Question 5

Tom says: "In general,

X – algorithm will NOT work correctly on a

Y – algorithm will NOT work correctly on a

Z – algorithm will NOT work correctly on a ."

How about you?

◯ Disagree with all 3	◯ Agree with X only	◯ Agree with Y only	◯ Agree with Z only
◯ Agree with X, Y only	◯ Agree with X, Z only	◯ Agree with Y, Z only	◯ Agree with X,Y,Z

Question 6

You want to implement your OWN

You are able to find a

.

Which of these statement(s), if any, are correct:

X – We can implement the A

Y – There is no need for , why bother ?

Z – We can fix the

◯ Disagree with all 3	◯ Agree with X only	◯ Agree with Y only	◯ Agree with Z only
◯ Agree with X, Y only	◯ Agree with X, Z only	◯ Agree with Y, Z only	◯ Agree with X,Y,Z

Question 7

You want to create your own

You design this for the day when your program is used by billions …

Which of these statement(s), if any, are correct:

X – If we use an

Y – If we use a

Z – If we use a

◯ Disagree with all 3	◯ Agree with X only	◯ Agree with Y only	◯ Agree with Z only
◯ Agree with X, Y only	◯ Agree with X, Z only	◯ Agree with Y, Z only	◯ Agree with X,Y,Z

Section 2 – Read the scenario carefully and use it to answer Questions 8-12 **[15 marks]**

There are **N** cards placed in a line. Each card
may be very large.
`cards`.

You are also given 5 **independent** problems, as well as 10 possible solutions summarized by their **main** algorithm and/or **main** data structure(s) besides those given to you. For each of **Q8-12**, choose:

(a) ONE solution summary that is the **most suitable and most efficient**

◯ S1	◯ S2	◯ S3	◯ S4	◯ S5
◯ S6	◯ S7	◯ S8	◯ S9	◯ S10

and

(b) best answer for the time complexity of the chosen solution summary

◯ O(log(log(N))	◯ O(log(N))	◯ O($(\log(N))^2$)	◯ 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(N^3)

Marking Scheme

- Any part (a) or part (b) with two or more options shaded will be treated as being completely wrong
- To discourage spamming of a solution, no **solution** (for part (a)) should be picked more than **once across questions** – If that happens, only the *highest mark* for that solution will be awarded *once* across the entire section
- The best answer for time complexity in part (b) may or may not be repeated across questions
- Each solution (for part (a)) is worth somewhere between 0-3 marks
- The choice of each part (b) will cause the awarded mark from part (a) to be scaled by a factor (i.e. multiplied), between 0.5 (completely wrong) and 1.0 (correct option with respect to *your* chosen ADT in part (a))

Solution Summaries

- S1: No data structure needed, algorithm involves
- S2: problem
- S3: problem
- S4: , algorithm
- S5: problem
- S6: problem
- S7: algorithm
- S8: algorithm
- S9: algorithm
- S10: algorithm

Question 8

Given `cards`, find and output the number of
card.

E.g. if `cards` = then the answer is 7, because there are :
1

Question 9

Given `cards`, find and output the that is
.

E.g. if `cards` = , then the answer is 5, which is the :
1

Question 10

You are given `cards` and a positive integer
you can
to.

Each
never increases. You have to stop when you
do not have

Find and output the

E.g. if `cards` = , then the answer is 5:
card with
card with
card with

E.g. if `cards` = , then the answer is 4:
card with
card with
card with
but unable to
needs to be

Question 11 – Looks similar to Q10 but different. Read carefully!

You are given `cards`. You start off
Each turn, you can

Each

You have to stop when

Find and output the smallest
such that you will be able to
times.

E.g. if `cards` = , then the answer is 4:
card with
card with
card with
card with
card with
card with

Question 12

You are given `cards` and also the
folded if:

- the
  or
- the first

can be **folded**

e.g. all cannot be folded
e.g. can be folded

Find and output, for each element in sequence, the
If there are multiple possible answers, just output any valid answer.

E.g. if `cards` = , then the answer is " ", because:
3 at
4 at
7 at
…

THIS IS A

BLANK

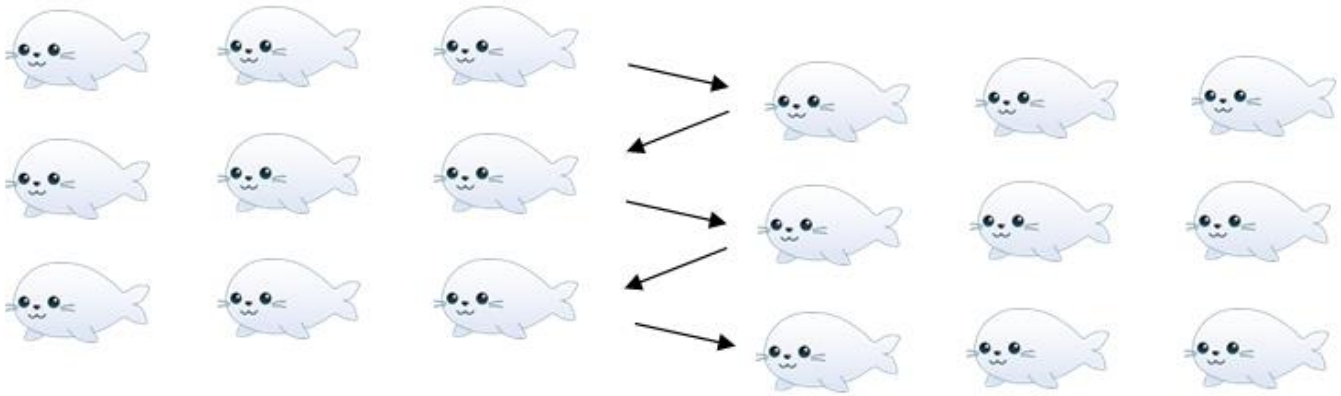
PAGE

Section 3 – Read the scenario carefully and use it to answer Questions 13-17

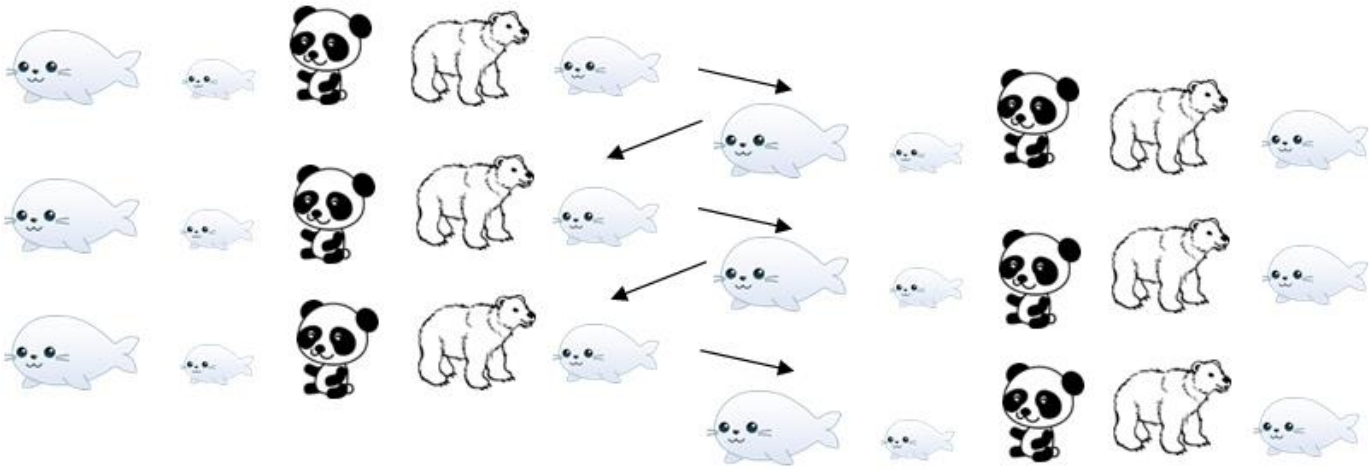
There are **N** animals[1] in a zoo
could possibly be very very large (but never grows). You are tasked to
using the

Each , you will be given a
but not more within that

As an **example**, for , , the required is **5**:



Another **example**, for , , the required is **5**:



[1] Images from http://clipart-library.com

Question 13 – Understanding the problem **[8 marks == 4 x 2]**

How many , for each of:

Q13a.

Q13b.

Q13c.

Q13d.

For each question in **Q13a-d**, choose the correct answer:

◯ 1	◯ 2	◯ 3	◯ 4	◯ 5	◯ 6	◯ 7	◯ 8	◯ 9	◯ 10

Question 14 – **[8 marks == 4 x 2]**

Ivan suggests that this problem is actually a **problem**. A the entire in which at least 1 after the

What are the ?

Q14a.	◯ Sharp	◯ Round
Q14b.	◯ Tall	◯ Short
Q14c.	◯ Far	◯ Near

Q14d. What is the most suitable above, if any?

◯ Black & White	◯ Grayscale	◯ Colour	◯ No image, what image?

Question 15 – **[10 marks == 5 x 2]**

Assume that there exists a

If there is a need to keep track of a hash map utilizing will be used instead of an . Assume that the be read in O(1) time, no matter how large it is.

Also assume that you **do NOT have control** over the

Which of these 4 algorithm(s) correctly solves the problem, ignoring efficiency? The problem may be solved by **AT LEAST 1** of these algorithms. For each of **Q15a-e**, shade the correct option:

◯ Yes	◯ No

Q15a. Run recursive

DO

Q15b. Run recursive

DON'T If

Q15c. Run

DO

Q15d. Run

DON'T

Q15e. Just for this part, assume that to simplify the analysis. Out of the algorithms you have **chosen among Q15a-d only**, what is the time complexity of the **best** algorithm?
[Reminder: ***N*** *is the number of animals and*
]

◯ O(**N**)	◯ O($\mathbf{N}^2$)	◯ O($\mathbf{N}^3$)	◯ O($\mathbf{N}^4$)	◯ Other worse polynomial time in **N**
◯ Exponential in **N**		◯ Factorial in **N**		◯ Could be worse than O(**N**!) even when…

Question 16 – Simulation **[6 marks == 3 x 2]**

Remember that is now not necessarily (back to just), and no longer a

There are more efficient algorithms to solve this problem. Suppose we **simulate** each using different algorithms/data structures to find the answer to the problem, which we will call **A**. In each only the best choice (or one of the many best choices if exist) will be made.

[**Reminder**: *As mentioned earlier, **A** could be very very large*]

For each of **Q16a-c** independent of one another, the algorithm correctly solves the problem. Choose the time complexity of using the respective algorithm to find the answer:

◯ O(**N**)	◯ O(**N** log**N**)	◯ O($\mathbf{N}^2$)	◯ O($\mathbf{N}^2$ log**N**)	◯ O(**AN**)	◯ O(**AN** log**N**)
◯ O(**A** $\mathbf{N}^2$)	◯ O(**A** $\mathbf{N}^2$ log**N**)	◯ O($\mathbf{A}^2$ **N**)	◯ O($\mathbf{A}^2$ **N**log**N**)	◯ O($\mathbf{A}^2\mathbf{N}^2$)	◯ O($\mathbf{A}^2\mathbf{N}^2$ log **N**)

Q16a.

While
    ?
    For
        D
Output

Q16b.

Create
Create
While
    For
        D
    E
    Clear
Output

Q16c.

Create
        each
Create
While
    Let
        Update
        Add
    Add
    Clear
Output

Question 17 – Most Efficient Algorithm **[1 mark (+3 bonus marks?)]**

Remember that is now not necessarily (back to just), and no longer a

Legend has it that there is still a more efficient algorithm to solve this problem!

[**Reminder**: ***N*** *is the number of animals. The time complexity should NOT depend on and/or* ***A*** *the answer to the problem]*

◯ O(log**N**)	◯ O($\sqrt{\mathbf{N}}$))	◯ O(**N**)	◯ O(**N** log**N**)
◯ O($\mathbf{N}^{1.5}$)	◯ O($\mathbf{N}^2$)	◯ O($\mathbf{N}^2$ log**N**)	◯ O($\mathbf{N}^3$)

[Bonus]

Implement the **most efficient** algorithm in **Java** for 3 **bonus** marks:

[**WARNING: Do not waste time on this part**, you will receive no credit at all for a solution that is incorrect or that is less efficient than the intended answer]

Box to answer is in answer sheet

- End of paper -