

# COL733 Project\_Exam

Viraj Agashe

TOTAL POINTS

**10 / 20**

## QUESTION 1

### Fugue 5 pts

1.1 Q 1.1 2.5 / 2.5

+ 0 pts Incorrect

✓ + 2.5 pts *Dense ordering is required to insert a character in between any two characters*

1.2 Q 1.2 2.5 / 2.5

+ 0 pts Incorrect

✓ + 2.5 pts *unique total order ensures consistent order of characters across replicas*

## QUESTION 2

### GFS with SWIM 5 pts

2.1 Q 2.1 0 / 2.5

✓ + 0 pts Incorrect

+ 2.5 pts master and CS were partitioned: CS was incorrectly declared dead

2.2 Q 2.2 0 / 2.5

✓ + 0 pts Incorrect

+ 2.5 pts One CS can talk to another CS but cannot talk to master

## QUESTION 3

### RedBlue Consistency 5 pts

3.1 Q 3.1 1 / 1

+ 0 pts Incorrect

✓ + 1 pts *Correct*

3.2 Q 3.2 2 / 2

+ 0 pts Incorrect

✓ + 2 pts *Divergent state*

+ 1 pts Missing example

3.3 Q 3.3 1 / 2

✓ + 0 pts Incorrect

+ 2 pts Make accrueInterest commute with deposit by changing multiplication to addition  
+ 1 Point adjustment

💬 Missed main point: make it commute

## QUESTION 4

4 Pregel 1 / 5

+ 0 pts Incorrect

+ 2 pts Vertices with no change vote to halt

+ 2 pts supersteps

✓ + 1 pts *Only active vertices send messages*

- 2 pts Computed max, not min.

I do hereby undertake that as a student at IIT Delhi:

(1) I will not give or receive aid in examinations, and

(2) I will do my share and take an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honour Code.

Each question is of 5 marks. Please attempt only 3 out of 4 questions. Do not attempt a question from your own project.

### Q1. Fugue

Vidushi, Chinmay, and Aryan implemented a collaborative editor with Fugue.

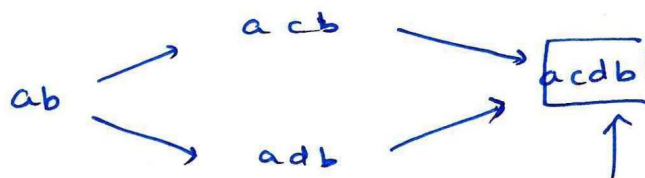
Q1.1 [2.5 marks] The group mentions in their slides that managing distributed indices requires a "dense ordering". What is meant by a dense ordering? Why is it important to have a dense ordering?

1.1 Dense ordering means that we can find an index  $i$  between ALL indices  $i_1, i_2$ , i.e.  $\forall i_1, i_2, \exists i$ .  
 $i_1 < i < i_2$

We need dense ordering as we need to keep indices <sup>of each char</sup> fixed, and we can add arbitrary no. of chars. between any 2 indices.

Q1.2 [2.5 marks] The group also mentions that indices need a unique total order. What goes wrong otherwise?

1.2 We need a unique total order so that when we merge 2 versions, we get a unique string.



If indices  $c, d$  are not totally ordered,

the string  $acdb$  will actually be  $acdb, adcb$  → conflicting versions

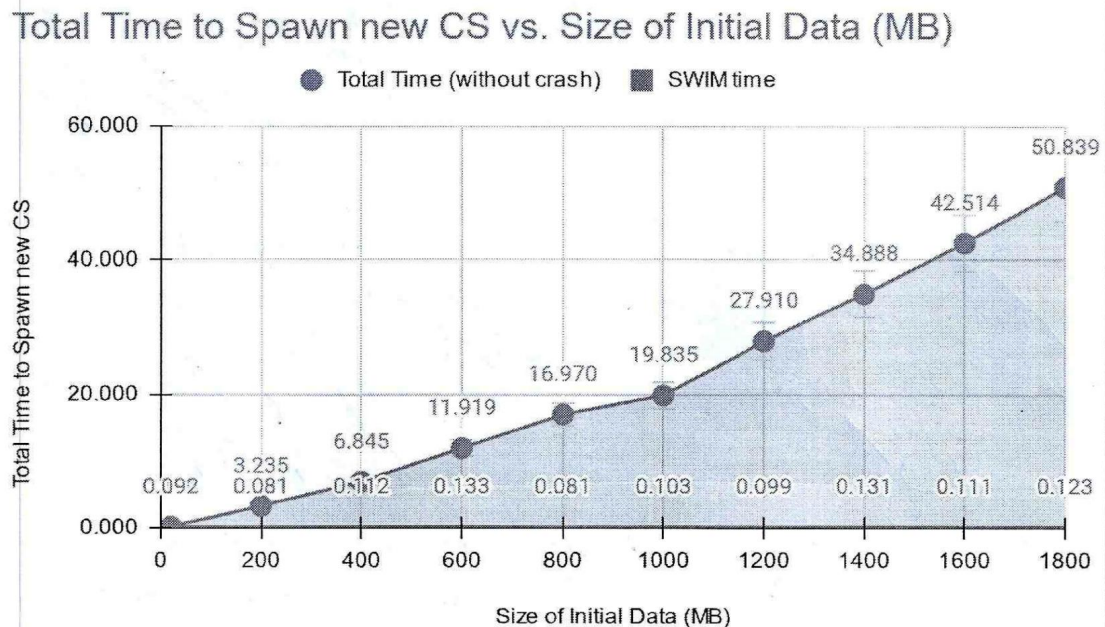
Name: \_\_\_\_\_

Entry number: \_\_\_\_\_

## Q2 GFS with SWIM

Reedam, Shivam, and Viraj replaced chunkserver failure detection in GFS with SWIM protocol.

The group reported the following chart in their presentation:



Q2.1 [2.5 marks] Explain the experimental setup for this figure. In particular, without SWIM why did we need to spawn a new chunkserver even though the chunkserver had not crashed?

Q2.2 [2.5 marks] With SWIM, why did we not need to spawn a new chunkserver in the above experiment? Explain with an example.

## Q3 RedBlue consistency

Satyam and Rohith implemented RedBlue consistency in Redis.

Q3.1 [1 mark] [True/False] Blue operations can be expected to see a lower latency than red operations.

TRUE .

Q3.2 [2 marks] The presentation described why `accrueInterest` and `deposit` can not be naively made blue operations. What goes wrong? Give an example.

If these are made blue ops naively, then since these do not commute, it leads to inconsistent state.

100  
↓ DEPOSIT (20)  
120  
↓ AI  
126

≠

100  
↓ AI  
105  
↓ DEPOSIT (20)  
125

Q3.3 [2 marks] How were these operations later made blue?

We created modified versions of these operations - shadow bank operations -

i.e. here we separate the decision of which transition to make from the act of applying the transition.

→ Changed to AI'(BAL \* INT).



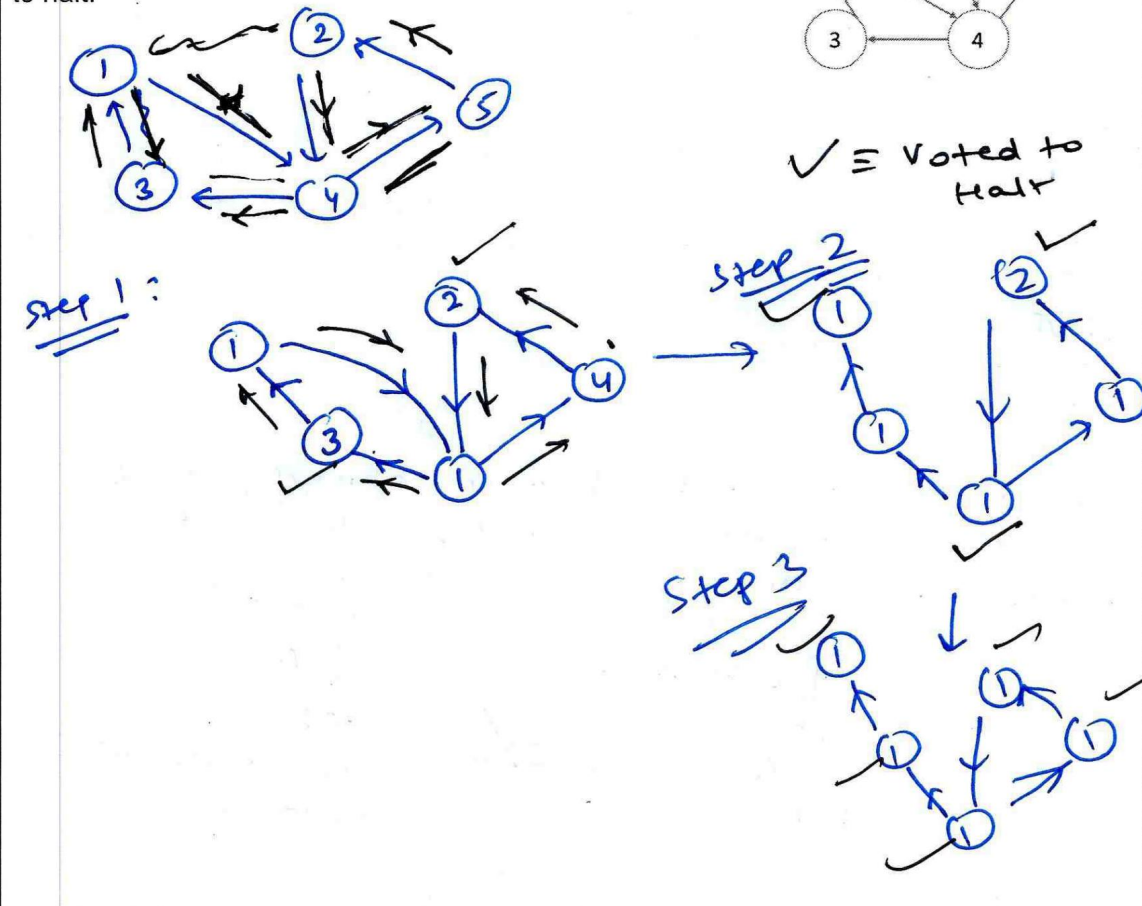
Name: \_\_\_\_\_

Entry number: \_\_\_\_\_

#### Q4 Pregel

Two groups: Shivam, Geetansh, Hemank and Rishi, Srijan, Spandana, implemented Pregel.

Let us say we are computing the *minimum* value in this fully connected graph using Pregel. Please draw the supersteps. In each superstep, draw the message passed on every vertex and grey the vertices that would have voted to halt.



This area is intentionally left blank. You can use it for rough work.