NATIONAL UNIVERSITY OF SINGAPORE

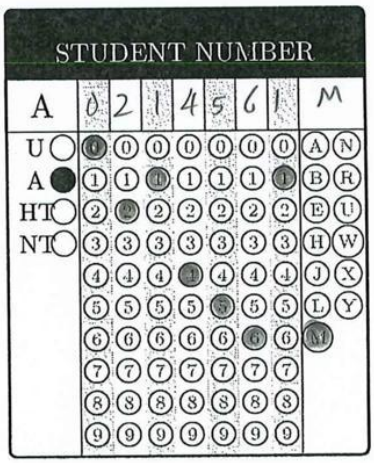
CS3223 - DATABASE SYSTEM IMPLEMENTATION

(Semester 2: AY2022/23)

Time Allowed: 100 minutes

INSTRUCTIONS TO STUDENTS

1. Write down your **student number** on the right and using ink or pencil, shade the corresponding circle in the grid for each digit or letter. DO NOT WRITE YOUR NAME!
2. The assessment paper contains **ELEVEN (11) questions** and comprises **ELEVEN (11) pages** including this cover page.
3. All questions must be answered in the space provided; no extra sheets will be accepted as answers. You may use the extra page behind this cover page if you need more space for your answers.
4. This is a **closed book** assessment.
5. You are allowed to refer to a single, double-sided A4-sized sheet of notes.
6. You are allowed to use an electronic calculator.
7. You are allowed to use pencils, ball-pens or fountain pens, as you like as long as it is legible (no red color, please).
8. **Marks may be deducted** for illegible handwriting



For Examiner's Use Only

Question	Marks
Q1	/ 2
Q2	/ 4
Q3	/ 2
Q4	/ 2
Q5	/ 2
Q6	/ 2
Q7	/ 2
Q8	/ 2
Q9	/ 3
Q10	/ 2
Q11	/ 2
Total	/ 25

This page is intentionally left blank.
Use it ONLY if you need extra space for your answers, in which case indicate the **question number clearly** on this page as well as in the original answer box.

Question 1: External Merge Sort [2 marks]

This question is about the external merge sort algorithm and it consists of **two independent parts.**

A. Suppose that you are given a file with **20,000 pages.**

Let N denote the number of initial sorted run produced using **10 buffer pages.**

Choose the most **appropriate** statement about N. [1 mark]

- ○ N = 1000
- ● N = 2000
- ○ N = 2500
- ○ N = 5000
- ○ N = 10000
- ○ N = 20000
- ○ All of the above statements are false.

1 / 1

B. Suppose that you are given **10,000 initial sorted runs** each of which is **10 pages** long. You have **10 buffer pages** available to merge these sorted runs into a single sorted run.

Consider the following buffer space allocation with 2 pages allocated for each input buffer and 2 pages allocated for the output buffer (i.e., blocking factor b = 2).

Let M denote the **number of merging passes** required to merge the initial sorted runs into a single sorted run based on this buffer space allocation.

Choose the most **appropriate** statement about M. [1 mark]

- ○ M = 1
- ○ M = 2
- ○ M = 3
- ○ M = 4
- ○ M = 5
- ○ M = 6
- ● M = 7
- ○ M = 8
- ○ M = 9
- ○ M = 10
- ○ M > 10

1 / 1

Question 2: Disk I/O [4 marks]

Consider a magnetic disk with the following specifications:

- Average seek time = s ms.
- Average rotational delay = r ms.
- Data transfer rate = d KB/ms.
- Disk block size = $8KB$.
- The disk can read from one head at a time.

Consider an application that needs to retrieve two data records from a file stored on this disk, where the size of each data record is $2KB$, and no record is allowed to span two disk blocks.

For each of the following two cases, write down an expression for the total access time to retrieve the two records from the disk.

2 / 2

A. The two records are stored on two distinct cylinders on the disk. [2 marks]

$\left(s+r+\frac{8}{d}\right) + \left(s+r+\frac{8}{d}\right) = \left(2s+2r+\frac{16}{d}\right)_{ms}$

B. The two records are stored on two consecutive disk blocks on the same track. [2 marks]

$\left(s+r+\frac{16}{d}\right)_{ms}$

2 / 2

Question 3: Query Evaluation: Join [2 marks]

Consider the evaluation of $R \bowtie S$ using the **block nested-loop join**, where $|R| = 20$ pages and $|S| = 24$ pages.

If the evaluation using **B** buffer pages results in the inner relation being scanned a total of **4 times**, select **all** the statements that are true about B. [2 marks]

- ○ B = 3
- ○ B = 4
- ○ B = 5
- ○ B = 6
- ● B = 7
- ● B = 8
- ○ B = 9
- ○ B = 10
- ○ All of the above statements are false.

2 / 2

R as outer
S is inner relation

not allowed to switch

If allow switch
S as outer
R as inner

B=9 also can $\left\lceil \frac{24}{9-2} \right\rceil = 4$

Question 4: Query Evaluation: Selection [2 marks]

Consider a relation with the schema **R (A, B, C, D, E)**, where attribute A is the primary key of R, and the following query **Q**:

SELECT * FROM R WHERE (B = 10) AND (C > 20) AND (D > 30).

Assume the following for this question:

1. Relation R contains 10,000 pages with each page containing 40 records.
2. I_{DB} is an unclustered B$^+$-tree index on (D,B) with 4000 leaf pages and 100 data entries in each leaf page.
3. I_{DB} has two levels of internal nodes.
4. 20,000 records in R satisfy the condition "B = 10".
5. 40,000 records in R satisfy the condition "C > 20".
6. 10,000 records in R satisfy the condition "D > 30".
7. 4,000 records in R satisfy the condition "(B = 10) and (C > 20)".
8. 5,000 records in R satisfy the condition "(B = 10) and (D > 30)".
9. 6,000 records in R satisfy the condition "(C > 20) and (D > 30)".
10. 2,000 records in R satisfy the condition "(B = 10) and (C > 20) and (D > 30)".

This question considers the evaluation of query Q using the index I_{DB}, and the question consists of **two parts**.

A. Let **L** denote the number of leaf pages in I_{DB} accessed by the evaluation of Q using I_{DB}.

Select the most **appropriate** statement about L. [1 mark]

- ○ L = 20.
- ○ L = 40.
- ○ L = 50.
- ○ L = 60.
- ◉ L = 100.
- ○ L = 200.
- ○ L = 400.
- ○ L = 4000.
- ○ All of the above statements are false.

1 / 1

B. Let **P** denote the (worst-case) number of data pages accessed in R by the evaluation of Q using I_{DB}.

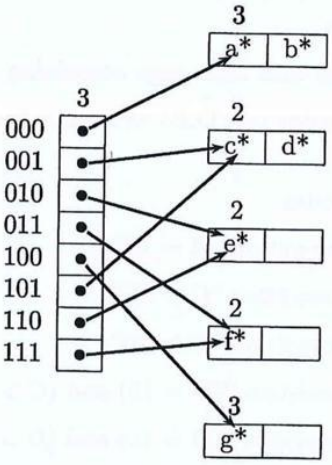
Select the most **appropriate** statement about P. [1 mark]

- ○ P = 2000.
- ○ P = 4000.
- ◉ P = 5000.
- ○ P = 6000.
- ○ P = 10000.
- ○ P = 20000.
- ○ P = 40000.
- ○ All of the above statements are false.

1 / 1

Question 5: Extendible Hashing [2 marks]

Consider the Extendible Hashing index I shown below where each data bucket page can store at most 2 data entries. Each letter shown (a to g) represents some integer value denoting the hashed value of some index key value.



Suppose the insertion of a new data entry $x*$ into I causes the directory to be doubled.

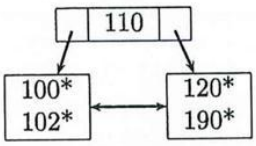
Select **all** the statements that are true about x. [2 marks]

- [x] The last three bits of x could be 000. ✓
- [x] The last three bits of x could be 001. ✓
- [ ] The last three bits of x could be 010. ✗
- [ ] The last three bits of x could be 011. ✗
- [ ] The last three bits of x could be 100. ✗
- [x] The last three bits of x could be 101. ✓
- [ ] The last three bits of x could be 110. ✗
- [ ] The last three bits of x could be 111. ✗
- [ ] All of the above statements are false.

2 / 2

Question 6: B$^+$-tree Index [2 marks]

Consider the B$^+$-tree index T with order $d = 1$ shown below.



In this question, assume that an overflowed internal/leaf node is always split (i.e., there is no redistribution of entries to resolve overflows). If an overflowed node N is split into N and N', then N contains d entries (where d is the order of the index) and N' is the right sibling of N.

Let T' denote the resultant B$^+$-tree index after inserting the following sequence of four data entries into T:

$$40^*,\ 70^*,\ 105^*,\ 103^*.$$

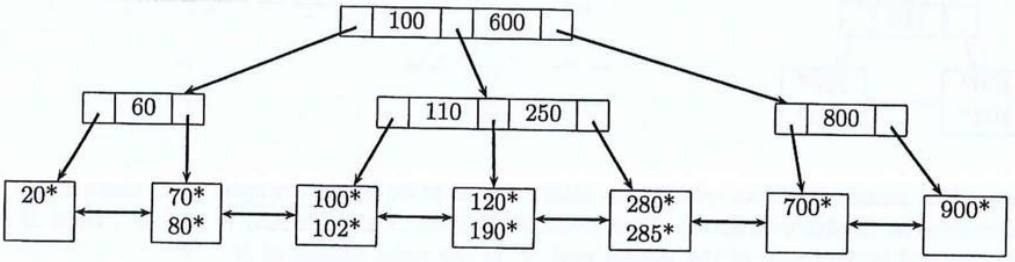
Select **all** the statements that are true about T'. [2 marks]

- ◯ The total number of internal nodes in T' is 1.
- ◯ The total number of internal nodes in T' is 2.
- ● The total number of internal nodes in T' is 3.
- ◯ The total number of internal nodes in T' is at least 4.
- ◯ The total number of leaf nodes in T' is 2.
- ◯ The total number of leaf nodes in T' is 3.
- ◯ The total number of leaf nodes in T' is 4.
- ● The total number of leaf nodes in T' is 5.
- ◯ The total number of leaf nodes in T' is 6.
- ◯ The total number of leaf nodes in T' is at least 7.
- ◯ There is a leaf node in T' consisting of only one data entry 103^*.
- ◯ There is a leaf node in T' with data entries 102^* and 103^*.
- ● There is a leaf node in T' with data entries 103^* and 105^*.

2 / 2

Question 7: B$^+$-tree Index [2 marks]

Consider the B$^+$-tree index T with order $d = 1$ shown below.



In this question, assume that an underflowed node N is always resolved by redistribution whenever possible. The redistribution policy is as follows: whenever possible, redistribute from the right sibling of N; otherwise, redistribute from the left sibling of N. If redistribution is not possible, then merge N with a sibling node as follows: if N has a right sibling, then merge N with its right sibling; otherwise, merge N with its left sibling.

Let T' denote the resultant B$^+$-tree index after deleting the data entry **700*** from T.

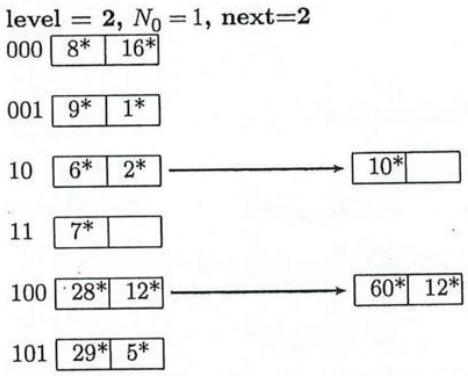
Select **all** the statements that are true about T'. [2 marks]

- ○ The total number of internal nodes in T' is 2.
- ○ The total number of internal nodes in T' is 3.
- ● The total number of internal nodes in T' is 4.
- ○ The total number of internal nodes in T' is at least 5.
- ○ The total number of leaf nodes in T' is 4.
- ○ The total number of leaf nodes in T' is 5.
- ● The total number of leaf nodes in T' is 6.
- ○ The total number of leaf nodes in T' is 7.
- ○ The total number of leaf nodes in T' is at least 8.
- ○ There is an internal node in T' with exactly one index entry for the key value 100.
- ● There is an internal node in T' with exactly one index entry for the key value 110.
- ○ There is an internal node in T' with exactly one index entry for the key value 250.
- ● There is an internal node in T' with exactly one index entry for the key value 600.
- ○ There is an internal node in T' with exactly one index entry for the key value 800.

2 / 2

Question 8: Linear Hashing [2 marks]

Consider the linear hashing index I with six buckets shown below. Assume that each bucket page can store at most two data entries.



Base-10	Base-2
1	000001
2	000010
4	000100
5	000101
6	000110
7	000111
8	001000
9	001001
10	001010
12	001100
13	001101
16	010000
25	011001
26	011010
28	011100
29	011101
60	111100

Assume that a bucket split is triggered whenever some bucket overflows.

Let $\mathbf{I}'$ denote the resultant linear hashing index after inserting into I the following sequence of four data entries: **13***, **4***, **26***, **25***.

Let **L** denote the value of `level` in I'.

Let **N** denote the value of `next` in I'.

Let $\mathbf{B}_i$ denote the number of buckets in I' that consists of exactly i data entries.

Select **all** the statements that are true about I'. [2 marks]

- ○ L = 2.
- ● L = 3. ✓
- ○ L = 4.
- ○ L = 5.
- ○ L = 6.
- ○ L > 6.
- ○ N = 0.
- ○ N = 1.

0 / 2

- ● N = 2. ✓
- ○ N = 3.
- ○ N = 4.
- ○ N > 4.
- ○ $B_2 = 0$.
- ● $B_2 = 1$. ✓
- ○ $B_2 = 2$.
- ○ $B_2 = 3$.
- ○ $B_2 = 4$.
- ○ $B_2 > 4$.
- ○ $B_3 = 0$.
- ○ $B_3 = 1$.
- ● $B_3 = 2$. ✗
- ○ $B_3 = 3$.
- ○ $B_3 = 4$.
- ○ $B_3 > 4$.

0221

Question 9: Extendible Hashing [3 marks]

Consider an Extendible Hashing Index I which is initially created with a global depth value of 0 and consists of one empty data bucket.

After inserting a sequence of data entries into I (without any deletions), let I' denote the resultant index and assume that the global depth of I' is 3.

This question consists of **three parts**.

A. Let B_{min} denote the minimum number of buckets in I'.

Choose the most **appropriate** statement about B_{min}. [1 mark]

- ○ $B_{min} = 1$
- ○ $B_{min} = 2$
- ○ $B_{min} = 3$
- ● $B_{min} = 4$
- ○ $B_{min} = 5$
- ○ $B_{min} = 6$
- ○ $B_{min} > 7$

1 / 1

B. Let E_{max} denote the maximum number of **empty buckets** in I'.

Choose the most **appropriate** statement about E_{max}. [1 mark]

- ○ $E_{max} = 0$
- ○ $E_{max} = 1$
- ● $B_{min} = 2$
- ○ $B_{min} = 3$
- ○ $B_{min} = 4$
- ○ $B_{min} = 5$
- ○ $B_{min} = 6$
- ○ $B_{min} > 6$

1 / 1

$B_{min} = E_{max}$

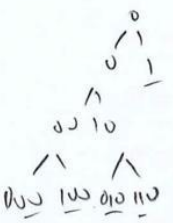
splitting back up

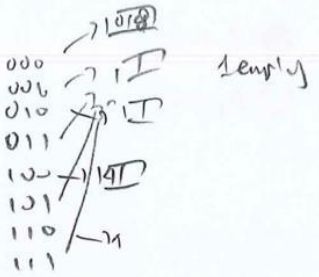
C. Let B_{max} denote the maximum number of buckets in I'.

Choose the most **appropriate** statement about B_{max} if there is at least one bucket in I' with a local depth of 1. [1 mark]

- ○ $B_{max} = 0$
- ○ $B_{max} = 1$
- ○ $B_{max} = 2$
- ○ $B_{max} = 3$
- ○ $B_{max} = 4$
- ● $B_{max} = 5$
- ○ $B_{max} = 6$
- ○ $B_{max} > 6$

1 / 1





1 empty

Question 10: Buffer Manager [2 marks]

Consider a buffer pool consisting of 6 frames (numbered 0 to 5) that is managed using the **clock replacement policy**. The following table shows the current state of the buffer pool, where each row indicates the pin count, referenced bit, and dirty flag values for a buffer frame.

Frame Number	Pin Count	Referenced Bit	Dirty Flag
0	0	off	true
1	0	off	false
2	0	on	true
3	1	off	false
4	2	off	true
5	0	on	false

Assume that the `current` variable is pointing to **buffer frame number 2**. Suppose that a victim page is to be selected for replacement. Let v denote the frame number of the victim page selected by the buffer manager for replacement. Choose the most **appropriate** statement above v. [2 marks]

- (●) v = 0.
- ( ) v = 1.
- ( ) v = 2.
- ( ) v = 3.
- ( ) v = 4.
- ( ) None of the buffer frames can be selected for replacement.

2 / 2

v=5

Question 11: Linear Hashing [2 marks]

Consider a Linear Hashing Index I with the following properties: (1) I consists of 4 data buckets, where all the buckets are non-empty; (2) each bucket page in I can store at most n data entries, where n ≥ 2; and (3) there are no overflow pages in I.

Consider the insertion of two data entries $x*$ and $y*$ into I.

There are two possible ways to perform these insertions.

The first way is to insert $x*$ followed by $y*$. Let I_x denote the resultant index formed by inserting $x*$ into I, and I_{xy} denote the resultant index formed by inserting $y*$ into I_x.

The second way is to insert $y*$ followed by $x*$. Let I_y denote the resultant index formed by inserting $y*$ into I, and I_{yx} denote the resultant index formed by inserting $x*$ into I_y.

For this question, assume that a bucket split is triggered whenever some bucket overflows.

Determine whether the following statement is true or false: If the creation of I_{xy} from I results in $x*$ being stored in an overflow page in I_{xy}, then the creation of I_{yx} from I must also result in $x*$ being stored in an overflow page in I_{yx}. [2 marks]

- (●) False
- ( ) True

2 / 2

— END OF PAPER —

0221