

[Home](#)[Grades](#)[View submissions](#)[View feedback](#)[Logout](#)

jmajikes

Assignment was due by 2021-11-09 15:30:00

## Midterm 2 COMP 421 Fall 2021

**Note:** There are a total of 100 points on this exams.

**Don't panic!**

You have 75 minutes to finish the exam.

- You *should* stay in full screen mode.
  - A <ctrl-f> find will give you a warning, which is OK
- You must hand this midterm in on time.
  - Points removed for late submissions.
- Absolutely no exams will be accepted after the end of class.
  - Only your **first** submission will be accepted/graded.
  - Avoid accidental submissions. Fill in your name when you are ready to submit.

I recommend that you have several pieces of scrap paper to doodle notes on during the exam. I *strongly* recommend you read the whole exam and begin with questions you know how to solve quickly. Some questions will be harder or take longer than others; don't spend all your time on one question worth only a few points! Also, don't worry if you don't have time to solve every question - that's expected.

Consider this midterm **closed book**.

You can **NOT** reference other online homeworks, worksheets, etc.

You can use your notes or other things printed out. They should be on paper as you may not switch screens after starting the exam.

You **MAY NOT** Google for anything, You **MAY NOT** leave this website, you **MAY NOT** visit any websites, and you **MAY NOT** copy from a friend. Do not paste information into your midterm unless you know it came from your midterm. You **MUST NOT** not receive help from anyone.

If you do not know the origin of material you should not paste it into this exam. All material pasted into this exam must originate from this exam. This



implies, but is not limited to, copying from previous assignments, copying from text messages, or copying from **any** website.

You **MUST** use the Google Chrome browser.

The instruction team will **not** answer questions about course content, SQL syntax, etc. We will only deal with issues related to exam implementation.

If your browser hangs, for example because of a bad SQL query, simply kill the page and refresh. It *should* restore all your work even if it doesn't re-evaluate all answers, color-highlight boxes, etc.

You may **NOT** leave before you submit your exam. When you submit your exam you **must enter the code** displayed on the screen at the front of the class or given to you by ARS. `

You **must not** use your computer or phone in the classroom after you submit your exam.

The browser will change input box color **green** to indicate correctness. A black or **red** box indicates an incorrect answer.

Note that HTML select statements with drop-downs are simple multiple choice questions. No highlighting of correct answers are done for select questions.

**Green** highlight should just assist you. If you believe your answer is correct and the input box did not turn **green**, continue on. Per the [syllabus](#), highlighting is simply an aide not a guarantee. **Note:** For database queries that are applied to two databases, two green lights are required to get any credit for the question.

## Election Database Schema

Here are the tables you'll find for the database used in the midterm. Your queries will be run against two versions of the database. One of the databases will be much smaller and only contain a subset of the information. You should not consider this database information to be accurate to the current administration.

```
CREATE TABLE IF NOT EXISTS States
    (abbrev char(2) PRIMARY KEY,
     statename char(20), -- May include territori
     population integer)
```

```
CREATE TABLE IF NOT EXISTS Politicians
    (bioid char(20),
     firstname char(20),
     lastname char(20),
     birthday date, -- YYYY-MM-DD format string
     gender char(1),
     PRIMARY KEY(bioid))

CREATE TABLE IF NOT EXISTS Terms
    (termid INTEGER PRIMARY KEY AUTOINCREMENT,
     termtype char(20), -- Type of term elected; rep, sen,
     startdate date,
     enddate date,
     party char(20), -- Political party affiliation
     how char(20), -- Different ways to get into an office
     bioid char(20),
     -- Presidents & vice president aren't elected from districts
     -- Senators aren't elected from districts
     district integer, -- Null for prez, viceprez, or senator
     state char(2), -- Null for prez and viceprez
     FOREIGN KEY(bioid) REFERENCES Politicians(bioid))''')
```



The following scratch space can be used to help develop and test queries against a database described above.

Execute

**NOTE:** The database has been attenuated. For example, this database only has the original 13 states in the States relation. The Politicians relation only has the politicians born since the last century.

---

[Questions](#) For a total of 100 points

## SQL Queries 20 points

In this section, you will write SQL queries for the president and congressman schema described at the beginning of the exam. Your queries will be tested immediately against two different databases. If your queries output matches the expected output, the displayed answers will be outlined in green. You need green, highlighted output from the query of both databases to get any credit. Your actual score will be determined when your query is tested against a different database but **two** green feedbacks should mean that you are on track to receive full credit.

**Number.1:** List the first names of politicians that occur more than six times. For example, there may be more than 6 politicians having the first name of John.

Execute 10 points Minimize Output

**Number.Elected.7:** Write a query that returns the state names between A and G inclusive (include Florida but not Georgia) and the number **minus 2** of politicians that were elected from the state in 2021.

For example, if Connecticut elected 5 politicians in 2021, the output of the number elected minus 2 would be 3.

If the state did not have **more than two politicians elected in 2021**, return a count of Null for that state.

Execute

10 points

Minimize Output

## Evaluations 38 points

For the following three questions, assume that each tuple of Sailors is 50 bytes long, that a page can hold 80 Sailors tuples, and there are 250 pages.

Assume that Sailors has a index on search key  $\langle \text{Sailors.sid} \rangle$  that takes 50 pages and that there is no buffering to reduce redundant I/O. Further assume that the Sailors.sid is uniformly distributed between 0 and 39999, inclusive.

**B.Tree.IOs.2.1:** Assume that the index is a  $B^+$ -tree of height  $h \leq 10$  and that the index file is clustered, and that the most selective access path may use or ignore the index, estimate the number of pages retrieved for  $\sigma_{\text{Sailors.sid} < 20000}(\text{Sailors})$ .

**NOTE:** If you use the index you may use the variable  $h$  in your expression.

2 points

**B.Tree.IOs.2.2:** Assume that the index is a  $B^+$ -tree of height  $h \leq 10$  and estimate the number of pages retrieved for  $\sigma_{\text{Sailors.sid} = 20000}(\text{Sailors})$ .

**NOTE:** If you use the index you may use the variable  $h$  in your expression.

2 points

**Hash.IOs.3.1:** Now, assume that the index is a hash instead of a  $B^+$  tree, that the index file is unclustered, and that the optimizer may choose to use or ignore the index, estimate the number of pages retrieved for  $\sigma_{\text{Sailors.sid} < 20000}(\text{Sailors})$ .

2 points

For these five questions, use the Sailors relation (B+ tree index of height  $h$ , 250 pages, and 20000 tuples) and use the Reserve relation with a hash index  $\langle \text{Reserves.sid} \rangle$  that, on average, takes  $1+H$  I/Os to get to the data. The Reserve relation has 350 pages and 35000 tuples. For these five questions examine the I/Os for join of the following query:

```
SELECT S.sname, R.day
  FROM Reserves R, Sailors S
 WHERE S.sid = R.sid
```

**Simple.Nested.R.1:** Using a simple nested loop join, Reserves as the outer loop, no buffer optimization, and because we're using SNL the sid index is not used, how many I/Os does it take to complete the join?

5 points

**Block.Nested.S.1:** Using a block nested loop join, Sailors as the outer loop, no buffer optimization, and because we're using BNL the sid index is not used, how many I/Os does it take to complete the join?

5 points

**Which.Outer.Block.1:** If a block nested loop is used, no buffer optimization, and because we're using BNL the sid index is not used, which relation should be the relation on the outer loop in order to minimize I/Os?



2 points

**Buffered.Block.Nested.R.1:** Using a block nested loop join, Reserves as the outer loop, and because we're using BNL the sid index is not used, and 177 pages of buffers to optimize the outer loop, how many I/Os does it take to complete the join?

10 points

**Index.Nested.R.1:** Using an index nested loop join, Reserves as the outer loop, the index on Sailors.sid is used, and no buffer optimization for Sailors or Reserves, how many I/Os does it take to complete the join?

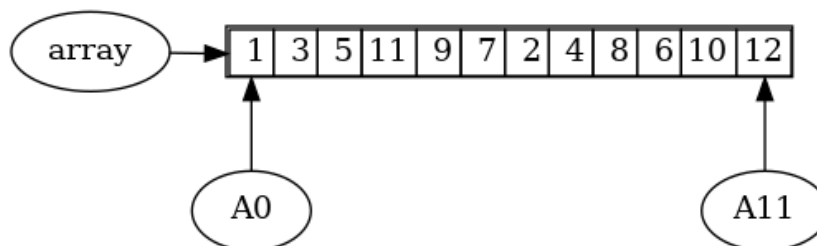
**Note:** This question gave a false-positive green light. It said that  $350 + (1 + h) * 20000$  was the correct answer.

You will still get a green light for that. But the correct answer is  $350 + (1 + h) * 35000$

Either will be accepted by the grader.

10 points

## External Sorting 20 points



**External.Merge.Sort.4.Walkthrough:** Given the above file of tuples with the keys shown, assuming that each page can hold two tuples, and using three buffers ( $B=3$ ), walk through the external merge sort.

If more rows appear in the table than passes required for the merge sort, simply repeat the sorted value to the extra rows.

Pass	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]	A[11]
Pass 0	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Pass 1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Pass 2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

5 points

**Calculator.4:** You may use this box as a calculator. Just type in any expression that can be evaluated by a JavaScript eval.

For example,  $2 ** 0.5$  will show you the result of the square root of 2 in the red box to the right. Leaving the box empty or filled in will **NOT** affect your grade.

**External.Merge.Sort.4.Table:** Suppose you have a file that has 100 pages that must be externally sorted and you have only 3 buffers.

**Note:** On the grade output and incorrect answer will appear as

Simple.Two.Way.Sort.4

Question	Answer
How many runs will be produced in Pass 0 (first pass)?	<input type="text"/>
How many passes will it take to sort the file completely? Remember, if it takes Pass 0, 1, 2 then that's 3 passes.	<input type="text"/>

2.5 points

**External.Merge.Sort.4.IOS:** Assuming one I/O per each page read and each page write, how many I/Os does it take to externally sort the 100 pages with 3 buffers?

2.5 points

**External.Merge.Sort.4.Two.Passes:** What is the minimum number of buffers required to sort the 100 pages in two passes?

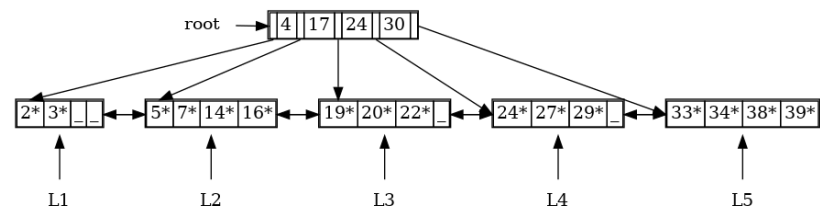
✓

10 points

**B Trees** 20 points

For these sets of queries assume the following B+ Tree:

1. Even though record keys are defined with an asterisk notation, the select pull downs will **not** include the asterisk!
2. If leaf or index entries are empty, leave the selection field blank
3. You must use the algorithm discussed in class. For example, do not do redistribution of nodes.



**Which.Leaf.15:** Which leaf data-node should the 15\* be inserted into?

✓

1 point

**Leaf.Nodes.15:** Enter the original leaf node

**Index.Nodes.15:** Enter the index after the insert of 15\*.

and, if required, the new leaf node after 15\* inserted.

**Note:**The select pull-down intentionally does not have all values between up to 41

Position	Leaf Node	Possible New Leaf Node	Position	Root Node	Index1	Index2
Entry 1	<div>✓</div>	<div>✓</div>	Entry 1	<div>✓</div>	<div>✓</div>	<div>✓</div>
Entry 2	<div>✓</div>	<div>✓</div>	Entry 2	<div>✓</div>	<div>✓</div>	<div>✓</div>
Entry 3	<div>✓</div>	<div>✓</div>	Entry 3	<div>✓</div>	<div>✓</div>	<div>✓</div>
Entry 4	<div>✓</div>	<div>✓</div>	Entry 4	<div>✓</div>	<div>✓</div>	<div>✓</div>

9 points

**Chapter Reading Review** 2 points

**Evaluate.Relational.Operators.1:** According to the book, which of the following techniques is commonly used to evaluate relational operators?



**Note:** Please read Chapter 12.2.1 page 398

1 point

**System.Catalog.2:** Which of the following database characteristics is required to be accurate in the system catalog in order to complete query evaluation?

**Note:** Please read Chapter 12.1.1 page 395

1 point

---