ECS 165B Spring 2010
Quiz #1 Name: _____

This is a closed book, closed notes quiz. Please turn cellphones off. You have 30 minutes. There are 14 problems worth 19 points in total (plus an extra credit problem worth 2 points).

Problem 1. (*1 point*) With variable-length records, which strategy is preferable for keeping track of free space in a heap file (aka Record File)?

- ☐ Linked list-based approach
- ◉ Directory-based approach

Problem 2. (*1 point*) In `DavisDB`, what components make up a `RecordID` structure?

- ☐ A file handle and an offset
- ☐ A bitmap and an array of record data
- ◉ A page number and a slot number

Problem 3. (*1 point*) Insertion of an entry into a full leaf page in a B+ tree results in

- ☐ Merging
- ◉ Splitting
- ☐ Redistribution

Problem 4. (*1 point*) Which kind of records does `DavisDB` use?

- ◉ Fixed-length records
- ☐ Variable-length records

Problem 5. (*1 point*) True or false: marking a page as dirty in `DavisDB` results in the page being written to disk immediately.

- ☐ True
- ◉ False

Problem 6. (*1 point*) Which advantages does a clustered B+ tree index have over a sorted file of records?

- ☐ Faster search *(1/2 point partial credit for checking this)*
- ☐ Faster updates *(1/2 point partial credit for checking this)*
- ◉ Both
- ☐ Neither

Problem 7. (*1 point*) Hash-based indices are useful for

- ◉ Equality search
- ☐ Range selection
- ☐ Both
- ☐ Neither

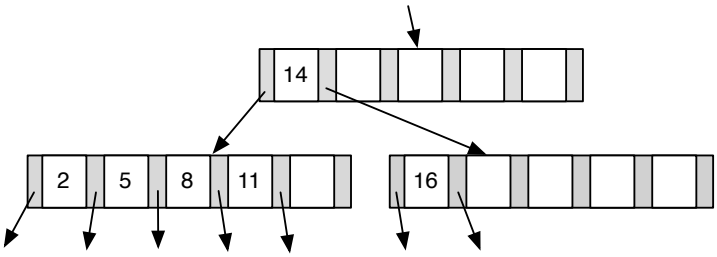
Problem 8. (*1 point*) What happens in the first pass (PASS 0) of an external merge sort?

- ☐ Pairs of runs are merged and written to disk
- ☐ A hash function h is used to create partitions
- ◉ Pages of records are read into memory, sorted in memory, and written to disk

Problem 9. (*1 point*) True or false: using an unclustered B+ tree index for sorting is usually faster than using external merge sort.

- ☐ True
- ◉ False

Problem 10. (*2 points*) In a B+ tree implemented using the "textbook" insertion and deletion algorithms described in lecture, why can't a tree like the one below (leaves not shown) occur?



Answer: The textbook algorithms guarantee a minimum occupancy of 50% for each node (except for the root), and here the node on the bottom right is less than 50% full.

Problem 11. (*2 points*) Consider a schema with two relations $R(A, B)$ and $S(C, D, E)$. Translate the following relational calculus query to SQL:

$$\{(x, y) \mid \exists z \; R(x, z) \wedge S(z, y, y)\}$$

Answer:

```
select R.A, S.D
from R, S
where R.B=S.C and S.D=S.E
```

or

```
select R.A, S.E
from R, S
where R.B=S.C and S.D=S.E
```

Problem 12. (*3 points*) Consider a schema with two relations $U(A, B)$ and $V(B, C)$, and suppose there are no indices on U or V. Name three algorithms that can be used to compute $R \bowtie S$.

Answer: Possible algorithms include (1) Simple Nested Loops Join, (2) Block Nested Loops Join, (3) Sort-Merge Join, and (4) Hash Join.

Problem 13. (*1 point*) Continuing with the schema from above, now suppose that V has an index on attribute B. Name a fourth algorithm that can now be used to compute $R \bowtie S$ (and which cannot be used without the index).

Answer: Possible algorithms include (1) Index Nested Loops Join, and (2) Sort-Merge Join (using the index to perform the sort).

Problem 14. (*2 points*) The following fragment of code in your `RecordFileHandle` implementation seems to be working incorrectly:

```
void RecordFileHandle::writePageHeader(char* data, int pageNoPrev,
     int pageNoNext, Bitmap* bitmap) {
   memcpy(data, &pageNoPrev, sizeof(pageNoPrev));
   memcpy(data+sizeof(pageNoPrev), &pageNoNext, sizeof(pageNoNext));
   memcpy(data+sizeof(pageNoPrev)+sizeof(pageNoNext), bitmap, sizeof(bitmap));
}
```

What is the likely problem?

Answer: The call to `sizeof` in the third `memcpy` returns the size of the *pointer* to the bitmap, rather than the size of the bitmap itself. Hence only part of the bitmap is copied into the `data` buffer.

Extra credit. (*2 points*) A volcano in Iceland erupted on April 14, disrupting air travel in Europe for nine days. Spell the name of that volcano correctly. (You may use extra pages if necessary.) Partial credit possible for creative answers.

Answer: Eyjafjallajökull