This assignment is designed to make you reflect on the problems that database research focuses on. It is better for you to write your own thoughts on questions 2, 3, 4, 5, and 7 than going online and checking for the best answers for your own development. You are free to do so after you come up with your own answer.

1) According to the given database schema above, write SQL queries and give the relational algebra queries of these questions.

   a) (5+5 points) Show full names (Fname, Minit, Lname) and SSNs of all the employees who don't have a supervisor.

   b) (5+5 points) Show full names (Fname, Minit, Lname) and SSNs of all the employees who don't work on any projects.
c) (5+5 points) How many hours does the employee who works on the highest number of projects work?

d) (10 points) If you were to put two indexes on each table (except DEPT_LOCATIONS), which columns would you choose on each table? Discuss for each table in a few sentences.

2) (10 points) Do you have to use indexes in the main memory? Defend your answer.

3) (10 points) Suppose you have a table that consists of 100 attributes. There are 100,000 records in this table. It is guaranteed that there is not going to be any insertions, updates or deletes to the table within the lifetime of the database. Would you prefer ISAM index or B+ tree index for this table? Defend your answer.

4) (10 points) Why do we need different join algorithms? Explain in your own words.

5) (10 points) Explain what is the best case, average case and worst case for quick sort, heap sort, merge sort, insertion sort and external sort (i.e., It is the worst case for quick sort when the list is ordered). If any two or three cases are the same, explain why.

6) Consider the B+ tree given in Figure 1. Using this tree, please answer the questions below individually (Not one after the other).

![Figure 1: B+ Tree](image)

a) (5 points) Insert 49.

b) (5 points) Insert 23.

c) (5 points) Insert 90.

d) (5 points) Delete 54.

7) (Bonus 10 points) Speculate on a problem (any problem) that the 6 join algorithms don't address. Some examples are parallelism, joining more than 2 relations at once, and encrypted databases. If you can think of anything else, you're welcome to explain it. Discuss how you can adapt one of the 6 algorithms (or a new algorithm if you can think of any) into solving the problem of your choosing.