1. (8 points) Consider the following SQL query.

```sql
SELECT *
FROM Heroes
WHERE hero_id > 399;
```

Assume a $B^+$-tree is defined on Heroes.hero_id. Describe how the index would be processed in order to answer the query. You may assume the tree has exactly two levels.
2. (2 points) What is the relational algebra operator that corresponds to the SELECT clause in an SQL statement?

3. (2 points) Given a table $R$, if the statistic $n_R$ must always equal $V(A, R)$, what must be true of attribute $A$?

4. (2 points) What is the most important cost portion of a query plan in a traditional DBMS?

5. (2 points) What are the TPC benchmarks?

6. (2 points) Define precision (as used with information recall).

7. (2 points) Define recall (as used with information recall).

8. (2 points) Define authorities (as used by the Google page rank algorithm).

9. (2 points) Define hubs (as used by the Google page rank algorithm).
10. (4 points) What does ACID stand for?

11. (4 points) What do the nodes and edges in a serialization graph represent?

12. (8 points) Create a 5NF decomposition over the relation \( R = (A, B, C, D, E) \) with only trivial functional dependencies.

13. (a) (2 points) Define the transitivity rule.

(b) (6 points) Using the definition of functional dependencies, prove the transitivity rule correct.
14. (8 points) Write an algorithm to find the $k$ nearest neighbors to a hero. The distance between two heroes is defined as the absolute value of the difference in the number of quests completed. Assume you have the lampquest database available.
15. (8 points) Provide the Hibernate mapping file for a simplified Hero object that has an id, a name, and a set of quests (which are also objects).

16. (8 points) Write a single SQL query to find for every quest, the quest name, the number of times the quest has been attempted, the number of times it has been completed and the most recent time it was attempted.
17. (8 points) Write a single SQL query to find the names of all of the heroes who have completed all of the quests.

18. (8 points) Write a single SQL query to find the names of each hero and all of the quests they have not completed.
19. (6 points) Draw an entity E with two attributes: a primary key K, and a composite attribute C consisting of a multivalued attribute M and a derived attribute D (you can ignore the note defining the derivation).

20. (6 points) In SQL, create all tables needed to instantiate E. Assume all attributes are strings.