Question 1: Sorting + Aggregation .... [130 points]

(i) **[10 points] Query Plan:**
Why are operators arranged in a tree representation?

(ii) **[10 points] Sorting:**
How is the `DISTINCT` clause implemented?

(iii) **[10 points] Sorting:**
Given a set of tuples, how can you accelerate loading into a B+Tree? Why?

(iv) **[10 points] Sorting:**
Is the quick-sort algorithm a good fit for sorting on-disk data? Why?

(v) **[10 points] Sorting:**
What are the number of passes required to sort a dataset with `N` pages?

(vi) **[10 points] Sorting:**
Explain how you could leverage additional buffer slots (`B` > 3) for sorting.

(vii) **[10 points] Sorting:**
List a data structure for efficiently compute the minimum element of a set of numbers.

(viii) **[10 points] Aggregation:**
Distinguish between two implementation choices for aggregation. When would you use either technique?

(ix) **[30 points] Aggregation:**
How is sorting similar to and different from hashing?

(x) **[10 points] Aggregation:**
Explain the external aggregation algorithm based on sorting.

(xi) **[10 points] Aggregation:**
Explain the external aggregation algorithm based on hashing.

End of Problem Set