1. As discussed in the lecture, the hardest part of projection operator is duplicate elimination. Briefly explain how sorting based approach and hash based approach work. What is the standard of these two approaches that is used for projection and why?

2. We have studied various join algorithms in class. Compare the following join algorithms, especially indicate their applicability and which algorithm works better under what situation.
   a. Block nested loop join
   b. Indexed nested loop join
   c. Sort merge join
   d. Grace hash join

3. Suppose there is relation Items I with C pages where C = 600 and each page holds 80 records, and relation Orders O with D pages where D = 2400 and each page holds 40 records. Answer the following questions (choose the order which results in fewer I/Os):
   a. I/O cost when performing nested loop join.
   b. I/O cost when performing block nested loop join with B = 220.
   c. I/O cost when performing sort merge join with B = 220.
Solution

1. Sorting based approach:
   - Modify Pass 0 of external sort to eliminate unwanted fields.
   - Modify merging passes to eliminate duplicates

Hashing based approach:
   - Partitioning phase: results into B-1 partitions (w/o) unwanted fields
   - Duplicate elimination phase: build in memory hashing function to further partition data and discarding duplicates.

The standard approach is sorting based approach because it is less impacted by skew and result is sorted.

2.
   a. Works for all join conditions, and always have the I/O cost [scan of outer + #outer blocks * scan of inner] no matter of the selectivity of the join conditions.
   b. Indexed nested loop join works better when the two joining tables are unbalanced, i.e., use the smaller table as outer table to index probe the inner table.
   c. Only works for equi-joins, if sorted was done previously this could be a huge advantage.
   d. Only works for equi-joins, also works better for unbalanced tables, i.e., build a hash table on small table, and scan the large table to do hash probing.

3.
   a. \( C + (C \times D) = 600 + (600 \times 2400) = 1440600 \) I/Os
   b. \( C + (\lceil C / B \rceil \times D) = 600 + (3 \times 2400) = 7800 \) I/Os
   c. Sorting I + Sorting O + C + D = 2400 + 9600 + 600 + 2400 = 15000 I/Os