The assignment is to be turned in before Midnight (by 11:59pm) on February 7, 2017. You should turn in the solutions to this assignment as a pdf file through the TEACH website. The solutions should be produced using editing software programs, such as LaTeX or Word, otherwise they will not be graded.

1: Query processing (2 points)

Consider the natural join of the relation R and S on attribute A. Neither relations have any indexes built on them. Assume that R and S have 40000 and 20000 blocks, respectively. The cost of a join is the number of its block I/Os accesses.

1. Assume that there are 50 buffer blocks available in the main memory. What is the fastest join algorithm to compute the join of R and S? What is the cost of this algorithm?

2. Assume that there are 300 buffer blocks available in the main memory. What is the cost of joining R and S using a sort-merge join? You should use a version of sort-merge algorithm that provides the minimum cost.

3. Assume that there are 202 buffer blocks available in the main memory. What is the cost of joining R and S using a sort-merge join? You should use a version of sort-merge algorithm that provides the minimum cost.

4. Assume that there are 150 buffer blocks available in the main memory. What is the fastest join algorithm to compute the join of R and S? What is the cost of this algorithm?

5. Assume that there are 80000 buffer blocks available in the main memory. What is the fastest join algorithm to compute the join of R and S? What is the cost of this algorithm?

2: Query processing (1 point)

Compare (improved) block-based nested-loop and sort-merge join algorithms for joining relations R and S in terms of their number of I/O accesses and memory requirements. Relations R and S do not fit in main memory. Your analysis should compare these algorithms for both the case where each tuple of R joins with a few tuples of S and the case in which each tuple of R joins with many tuples of S.

3: Query optimization (1 points)

Consider the following relations:

Product (name, production-year, rating, company-name)
Company (name, state, employee-num)

Assume each product is produced by just one company, whose name is mentioned in the company-name attribute of the Product relation. Attributes name are the primary key for relations Product and Company. Attribute company-name is a foreign key from relation Product to relation Company. Attribute rating shows how popular a product is and its values are between 1-5. The following statistics are available about the relations.
The following query returns the products with rating of 5 that are produced after 2000 and the states of their companies.

```
SELECT p.name, c.state
FROM Product p, Company c
WHERE p.company-name = c.name and p.production-year > 2000 and p.rating = 5
```

Suggest an optimized logical query plan for the above query. Then, estimate the size of each intermediate relation in your query plan. By an intermediate relation, we mean the relation created after each selection or join.