Assignment 1

1. Assume (for simplicity in this exercise) that only one tuple fits in a block and memory holds at most 3 page frames. Show the runs created on each pass of the sort-merge algorithm, when applied to sort the following tuples on the first attribute: (kangaroo, 17), (wallaby, 21), (emu, 1), (wombat, 13), (platypus, 3), (lion, 8), (warthog, 4), (zebra, 11), (meerkat, 6), (hyena, 9), (hornbill, 2), (baboon,12).

2. Let relations r1(A,B,C) and r2(C,D,E) have the following properties: r1 has 20,000 tuples, r2 has 45,000 tuples, 25 tuples of r1 fit on one block, and 30 tuples of r2 fit on one block. Estimate the number of block accesses required, using each of the following join strategies for r1⋈r2:

a. Nested-loop join
b. Block nested-loop join
c. Merge Join
d. Hash join

3. Suppose that a B+-tree index on building is available on relation department, and that no other index is available. What would be the best way to handle the following selections that involve negation?

i). σ ¬(building <"Watson")(department)

ii). σ ¬(building ="Watson")(department)

iii) σ ¬(building <"Watson" ∨ budget <50000)(department)

4. Consider the following Company Database relation and the SQL query, where the primary keys are underlined.

Employee(ssn,fname,lname,address,gender,salary)
Dependent(essn,dependent_name, gender)

```
SQL: Select A.fname, A.lname
from Employee A
where A.ssn IN (SELECT essn
                FROM Dependent
               WHERE essn=A.ssn and dependent_name=A.fname and gender=A.gender)
```

Write an efficient relational algebra expression that is equivalent to this query

5. Suppose that a B+-tree index on *branch-city* is available on relation *branch*, and that no other index is available. List different ways to handle the following selections that involve negation?

i) σ¬(branch-city<"Brooklyn")(branch)

ii) σ¬(branch-city<"Brooklyn" ∨ assets<5000)(branch)