1. Consider the relations $r_1(A,B,C)$, $r_2(C, D, E)$, and $r_3(E, F)$ in which r_1 has 1,000 tuples, r_2 has 1,500 tuples, and r_3 has 750 tuples.

a) Suppose the primary keys of r_1, r_2 and r_3 are A, C, and E, respectively. Estimate the size of $r_1 \bowtie r_2 \bowtie r_3$.

b) Assume that there are no primary keys, except the entire schema. Let $V(C, r_1)$ be 900, $V(C, r_2)$ be 1100, $V(E, r_2)$ be 50, and $V(E, r_3)$ be 100. Estimate the size of $r_1 \bowtie r_2 \bowtie r_3$.

a) The relation resulting from the join of r_1, r_2, and r_3 will be the same no matter which way we join them, due to the associative and commutative properties of joins. So, we will consider the size based on the strategy of $((r_1 \bowtie r_2) \bowtie r_3)$. Joining r_1 with r_2 will yield a relation of at most 1000 tuples, since C is a key for r_2. Likewise, joining that result with r_3 will yield a relation of at most 1000 tuples because E is a key for r_3. Therefore the final relation will have at most 1000 tuples.

b) The estimated size of the relation can be determined by calculating the average number of tuples which would be joined with each tuple of the second relation. In this case, for each tuple in r_1, $1500/V(C, r_2) = 15/11$ tuples (on the average) of r_2 would join with it. The intermediate relation would have 15000/11 tuples. This relation is joined with r_3 to yield a result of approximately 10,227 tuples ($15000/11 \times 750/100 = 10227$).

2. Consider the following schema, where the keys are underlined:

ENGINEER (ID, Name, Salary)
PROJECT (PID, ICEngID, Budget)

The ICEngID attribute in PROJECT is the ID of the engineer in charge of the project. The PID is the ID of the project. Both are sequential files in which records are stored in primary-key order. Consider the query


```
SELECT   *
FROM     ENGINEER E, PROJECT P
WHERE    E.ID=P.ICEngID AND P.Budget > 30
```

a) Write an unoptimized relational expression that might initially generate from the SQL query translator.

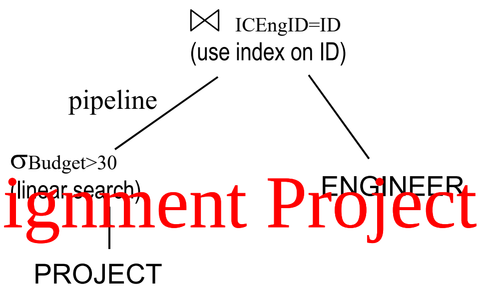
$\sigma_{Budget > 30}$ (PROJECT $\bowtie_{ICEngID=ID}$ ENGINEER)

b) Write an equivalent expression by fully pushing the selection.

($\sigma_{Budget > 30}$ (PROJECT)) $\bowtie_{ICEngID=ID}$ ENGINEER

c) Base on b) and use the following information, suggest an evaluation plan and estimate the cost in number of disk block transfers (ignore seek time here).

- M =5
- n_{ENGINEER} = 10,000
- b_{ENGINEER} = 2,000
- n_{PROJECT} = 2,000
- b_{PROJECT} = 500
- *min* (Budget, PROJECT) = 10
- *max* (Budget, PROJECT) = 60
- 4-level primary B^+-tree index on ID for ENGINEER
- 2-level secondary B^+-tree index on Budget for PROJECT



- Number of tuples satisfying the selection condition = 2000 * (60-30)/(60-10) = 1200
- Cost of selection = **500**
- Pipeline the selection output to the join operator.
- Cost of indexed nested loop join of the output of selection and ENGINEER = 1200 * (4+1) = **6000**
- Total cost = 500 + 6000 = **6500**
- Memory allocation: at the same time, 1 block for selection input (PROJECT), 1 block for selection output, 1 block for ID index / join input (ENGINEER), 1 block for join output