15.2

> Consider the bank database of Figure 15.14, where the primary keys are underlined, and the following SQL query:
>
> ```sql
> SELECT T.branch_name
> FROM branch T, branch S
> WHERE T.assets > S.assets AND S.branch_city = "Brooklyn"
> ```
>
> Write an efficient relational-algebra expression that is equivalent to this query. Justify your choice.

$\Pi_{Q.branch\_name}\left(\left(\rho_Q(branch) \bowtie_{T.assets>S.assets} \sigma_{branch\_city='Brooklyn'}(\rho_S(branch))\right) \bowtie \rho_T(bran$

15.3

> Let relations $r_1(A, B, C)$ and $r_2(C, D, E)$ have the following properties:
> r_1 has $20,000$ tuples, r_2 has $45,000$ tuples, 25 tuples of r_1 fit on
> one block, and 30 tuples of r_2 fit on one block. Estimate the number of
> block transfers and seeks required using each of the following join strategies
> for $r_1 \bowtie r_2$:
>
> a. Nested-loop join.
>
> b. Block nested-loop join.
>
> c. Merge join.
>
> d. Hash join.

a.

- r_1 作为外部关系：$30,000,800$ 次磁盘访问，$20,800$ 次磁盘寻道。
- r_2 作为外部关系：$36,001,500$ 次磁盘访问，$46,500$ 次磁盘寻道。

b.

- r_1 作为外部关系：$\lceil \frac{800}{M-2} \rceil \times 1500 + 800$ 次磁盘访问，$2 \times \lceil \frac{800}{M-2} \rceil$ 次磁盘寻道。
- r_2 作为外部关系：$\lceil \frac{1500}{M-2} \rceil \times 800 + 1500$ 次磁盘访问，$2\lceil \frac{1500}{M-2} \rceil$ 次磁盘寻道。

c.

假设 r_1 和 r_2 未排序，总排序成本为：

$$B_s = 1500 \times (2\lceil \log_{M-1}(1500/M) \rceil + 2) + 800 \times (2\lceil \log_{M-1}(800/M) \rceil + 2)$$

总成本为 $B_s + 1500 + 800$ 次磁盘访问。

d.

假设没有溢出，r_1 作为构建关系：

- 如果 $M > 800/M$，成本为 $3 \times (1500 + 800) = 6900$ 次磁盘访问。
- 否则，成本为：

$$2 \times (1500 + 800)\lceil \log_{M-1}(800) - 1 \rceil + 1500 + 800$$

次磁盘访问。