Physical Operators

Task

(refer to the handouts for the full exercise)

- What is the cost (in disk I/O's) of computing $R \bowtie_{R.A=S.B} S$ using the tuple-based nested loop join?

Physical Operators

Task

(refer to the handouts for the full exercise)

- What is the cost (in disk I/O's) of computing $R \bowtie_{R.A=S.B} S$ using the tuple-based nested loop join?

$$B(S) + B(R) \times B(S) = 200200$$

  This is the cost of reading S block by block, and matching the corresponding tuples for each block of R. Had we chosen R as the outer relation, the cost would have been higher.

- What is the minimum number of buffer pages required for this cost to remain unchanged?

  2

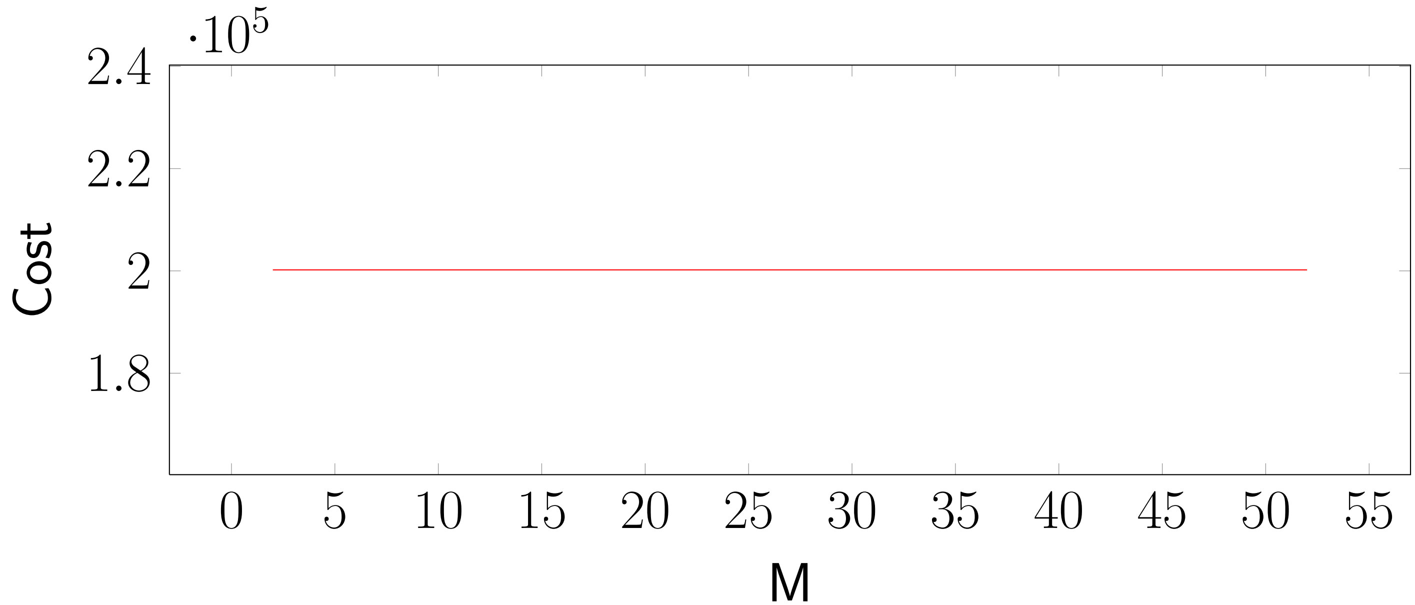
Physical Operators

Task

(refer to the handouts for the full exercise)

- What is the cost (in disk I/O's) of computing $R \bowtie_{R.A=S.B} S$ using the tuple-based nested loop join?

$$B(S) + B(R) \times B(S) = 200200$$



Physical Operators

Task

(refer to the handouts for the full exercise)

- What is the cost (in disk I/O's) of computing $R \bowtie_{R.A=S.B} S$ using the block-based nested loop join?

$$B(S) + B(R) \times \left\lceil \frac{B(S)}{M-1} \right\rceil = 4200$$

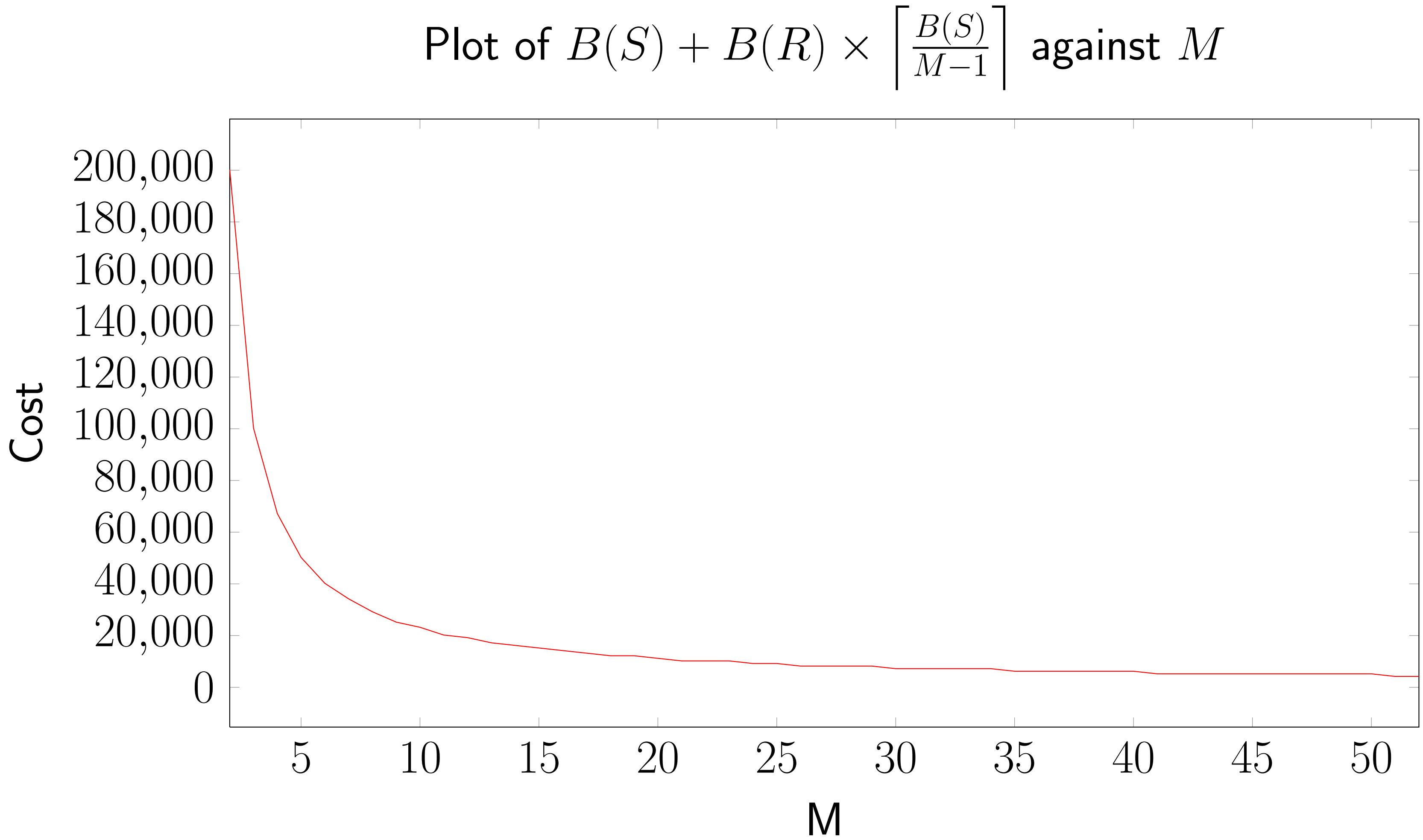
  If we consider R to be the "outer" relation, the cost will be higher.

- What is the minimum number of buffer pages required for this cost to remain unchanged?

$$51$$

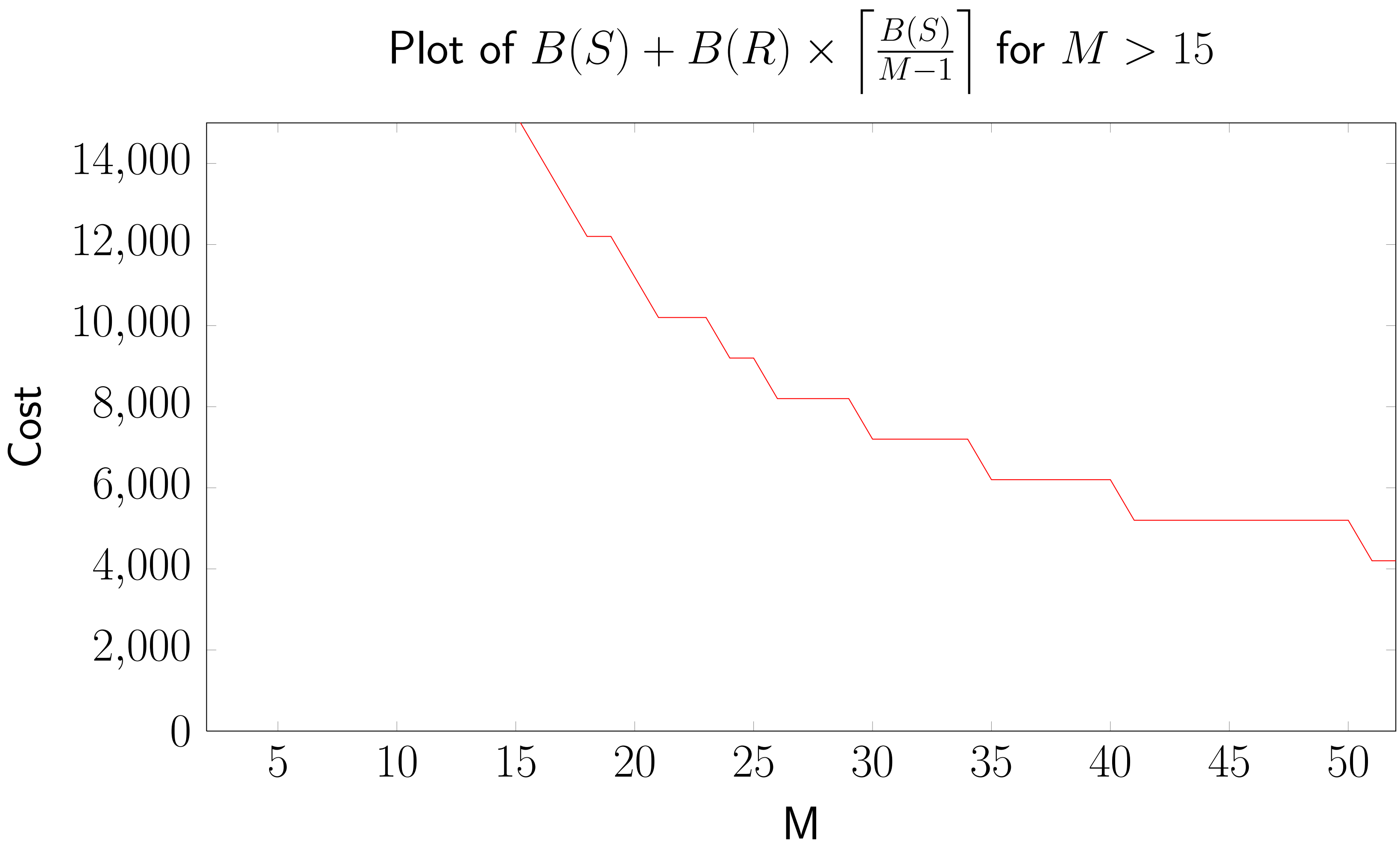
Physical Operators

Task



Physical Operators

Task



Physical Operators

Task

(refer to the handouts for the full exercise)

- What is the cost (in disk I/O's) of performing $R \bowtie_{R.A=S.B} S$ using a sort-merge join?

Physical Operators

Task

(refer to the handouts for the full exercise)

- What is the cost (in disk I/O's) of performing $R \bowtie_{R.A=S.B} S$ using a sort-merge join?

$$
\begin{aligned}
& 2B(R) \lceil \log_M B(R) \rceil + 2B(S) \lceil \log_M B(S) \rceil - B(R) - B(S) \\
= {} & 2 \times 1000 \times 2 + 2 \times 200 \times 2 - 1000 - 200 \\
= {} & 3600
\end{aligned}
$$

- What is the minimum number of buffer pages required for this cost to remain unchanged?

  The aforementioned join is computed in two passes. After one sorting pass, we can merge the whole relations:

$$\frac{B(R)}{M} + \frac{B(S)}{M} \leq M$$

  The minimal M that satisfies this is:

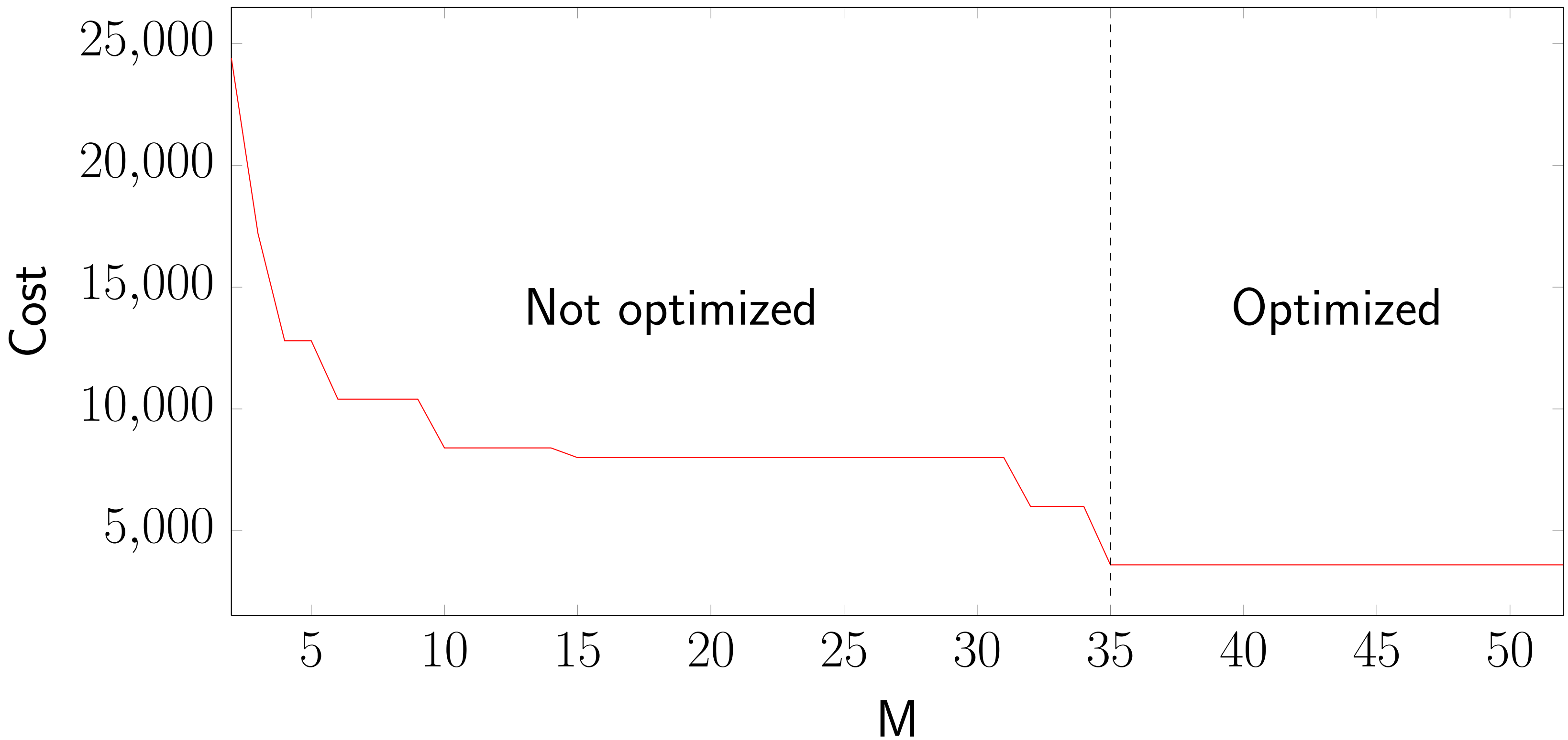
$$35 = \sqrt{B(R) + B(S)}$$

Physical Operators

Task

Plot of the I/O cost of the sort merge join as a function of M

$$\begin{cases} 2B(R)\lceil \log_M B(R)\rceil + 2B(S)\lceil \log_M B(S)\rceil - B(R) - B(S), & M >= 35 \\ 2B(R)\lceil \log_M B(R)\rceil + 2B(S)\lceil \log_M B(S)\rceil + B(R) + B(S), & M < 35 \end{cases}$$



Physical Operators

Task

(refer to the handouts for the full exercise)

- What is the cost (in disk I/O's) of performing $R \bowtie_{R.A=S.B} S$ using a hash join?

Physical Operators

Task

(refer to the handouts for the full exercise)

- What is the cost (in disk I/O's) of performing $R \bowtie_{R.A=S.B} S$ using a hash join?

$$\begin{aligned} & 2B(R)\left\lceil \log_{M-1} B(S) - 1 \right\rceil + 2B(S)\left\lceil \log_{M-1} B(S) - 1 \right\rceil + B(R) + B(S) \\ = & \; 2 \times 1000 \times 1 + 2 \times 200 \times 1 + 1000 + 200 \\ = & \; 3600 \end{aligned}$$

- What is the minimum number of buffer pages required for this cost to remain unchanged?
  The aforementioned hash join is performed in 2 passes. After one pass of hashing S, we must have buckets of a size that is at most $M-1$:

$$\frac{B(S)}{M-1} \leq M - 1 \Leftrightarrow M^2 - 2M + 1 - B(S) = 0.$$

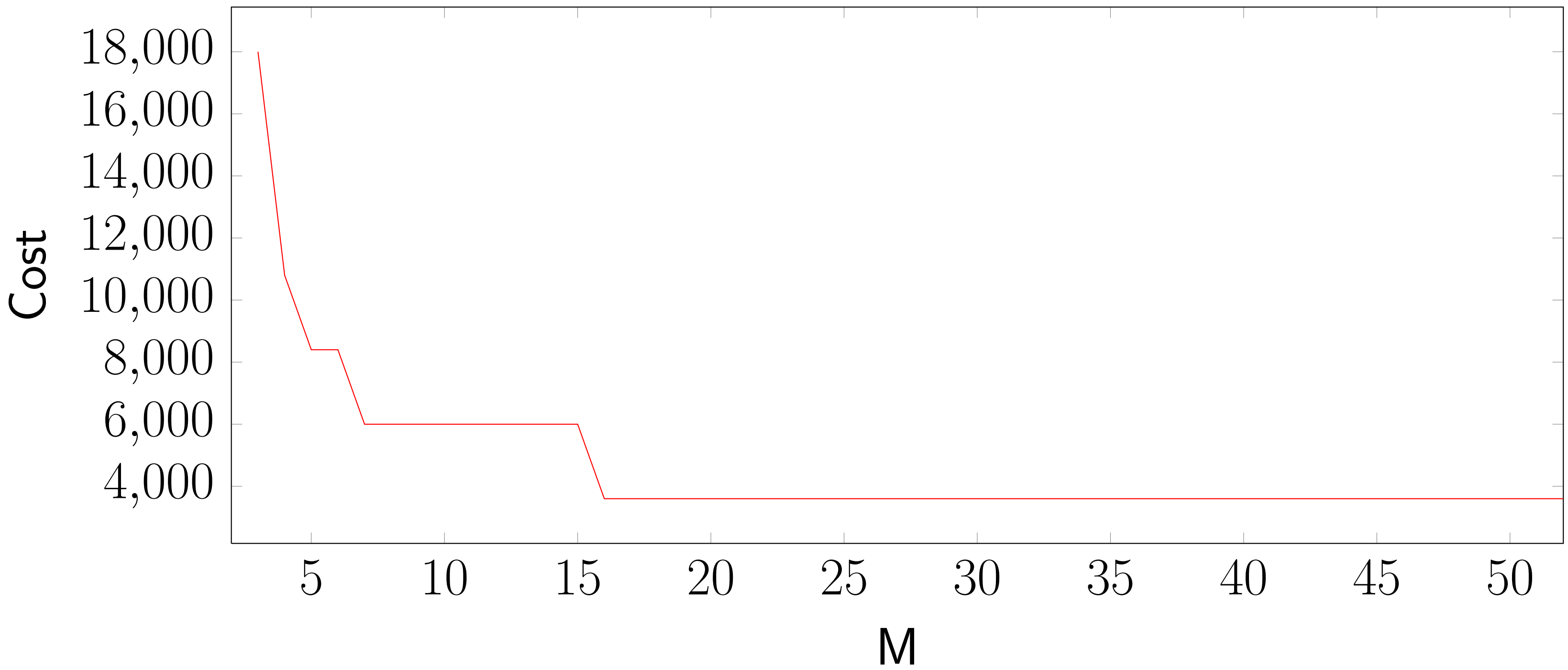
  The minimal M that satisfies is (positive root of the quadratic polynom): $\left\lceil 1 + \sqrt{200} \right\rceil = 16$.

Physical Operators

Task

Plot of the cost of the hash-based join as a function of M

$$2B(R)\left\lceil \log_{M-1} B(S) - 1 \right\rceil + 2B(S)\left\lceil \log_{M-1} B(S) - 1 \right\rceil + B(R) + B(S)$$



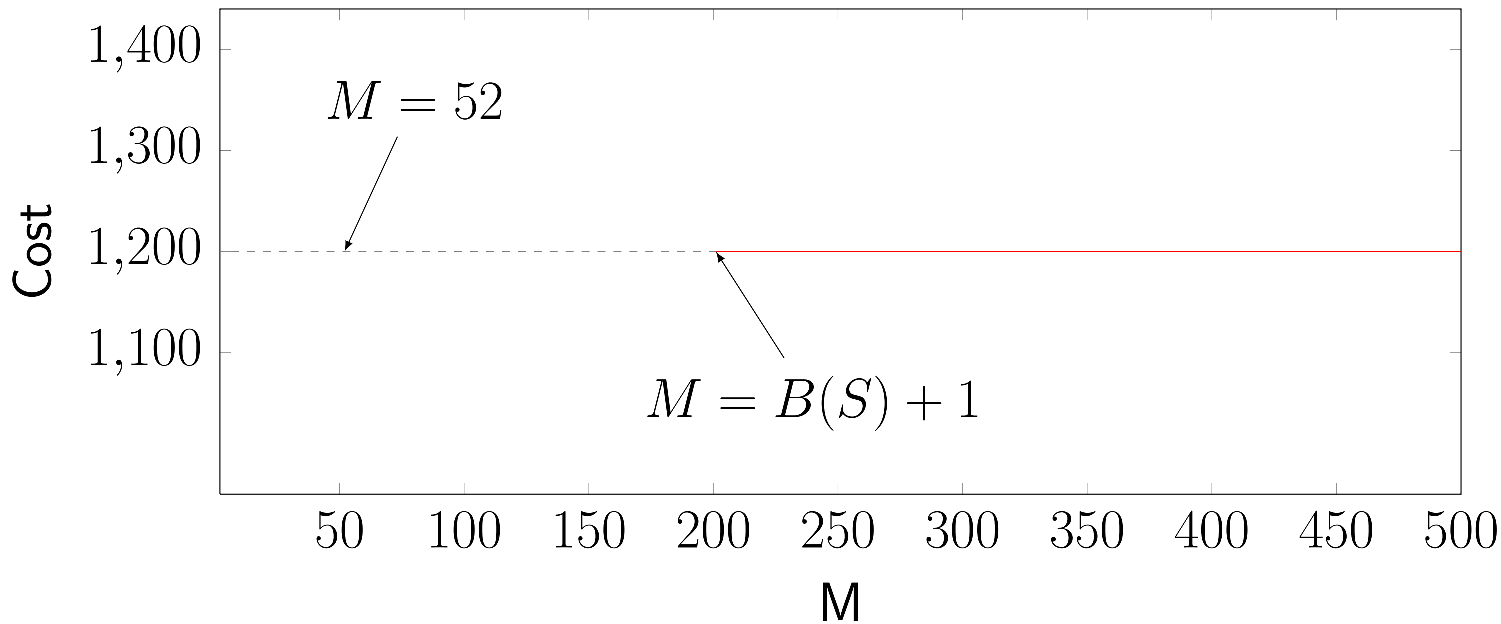
Note that we need at least 3 buffers (why?)

Physical Operators

Task

(refer to the handouts for the full exercise)

- What join algorithm yields the least cost if you were free to choose the number of free buffers?
  The single pass join. But You need $\min(B(R), B(S)) + 1$ buffers (why?)



Physical Operators

Task

(refer to the handouts for the full exercise)

How many tuples does the join of R and S produce, at most, and how many blocks are required to store the result of the join back on disk?

Since B is a key of relation S, there is at most one matching tuple in S for each tuple in R. Hence, the maximal number of tuples in the join is $T(R) = 10000$.

A resulting tuple is at most as large as a tuple of R plus a tuple of S. Hence, we have at most 5 of these tuples per block. The maximal number of blocks in the output is therefore 2000