Consider the join $R \bowtie S$, given the following information about the relations to be joined. The cost metric is the number of page I/Os, and the cost of writing out the result should be uniformly ignored.

Relation R contains 10,000 tuples and has 10 tuples per page.
Relation S contains 2000 tuples and also has 10 tuples per page.
Attribute $b$ of relation S is the primary key for S.
Each tuple in R will match exactly one tuple in S.
52 buffer pages are available.

1. What is the cost of joining R and S using a block nested loop join?
2. (a) With 52 buffer pages, if unclustered B+ indexes exist on $R.a$ and $S.b$, would either provide a cheaper alternative for performing the join (using an index nested loops join) than a block nested loop join? Explain. (Assume that it takes 3 I/Os to access a leaf in R, and 2 I/Os to access a leaf in S. And since S.b is a primary key, we will assume that every tuple in S matches 5 tuples in R.)
   (b) Would your answer to (a) change if only five buffer pages were available?
   (c) Would your answer to (a) change if S contained only 10 tuples instead of 2000 tuples? (Assume now all of the S tuples fit on a single page, and it will only require a single I/O to access the (single) leaf in the index. Also, each tuple in S will match 1,000 tuples in R.)
3. (a) With 52 buffer pages, if clustered B+ indexes existed on $R.a$ and $S.b$, would either provide a cheaper alternative for performing the join (using an index nested loops join) than a block nested loops join? Explain. (same assumption as 2.(a))
   (b) Would your answer to (a) change if only five buffer pages were available?
   (c) Would your answer to (a) change if S contained only 10 tuples instead of 2000 tuples? (same assumption as 2.(c))
4. If only 15 buffers were available, what would be the cost of a sort-merge join?
5. If the size of S were increased to also be 10,000 tuples, but only 15 buffer pages were available, what would be the cost of a sort-merge join?
6. If the size of S were increased to also be 10,000 tuples, and 52 buffer pages were available, what would be the cost of hash join?
11.2 Selectivity Estimation

Suppose we have two relations R(x,y) and S(y,z). It is also known that T(R) = 52, T(S) = 78 and V(R,y) = V(S,y) = 20, where T(R) is the number of tuples of relation R and V(R,A) is the number of distinct values of attribute A in relation R. Furthermore, min(R,y) = min(S,y) = 0 and max(R,y) = max(S,y) = 100. Consider the following expressions:

- $\sigma_{y=2}(R)$
- $\sigma_{y<3}(R)$
- $R \bowtie S$

1. If we assume uniform distribution, what is the selectivity of each of the expressions?

2. If the database holds a histogram of the two relations as follows, what are the selectivities?\(^1\)

<table>
<thead>
<tr>
<th>y</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>count (R)</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>count (S)</td>
<td>10</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>7</td>
<td>48</td>
</tr>
</tbody>
</table>

3. What are the error rates\(^2\) of the first estimation against the second?

11.3 Cost Estimation

Consider the query $\pi_{A,B,C,D}(R \bowtie_{A=C} S)$. Suppose that the projection routine is based on the modified external merge sort algorithm. Finally, assume that you know the following:

- R is 10 pages long, and R tuples are 300 bytes long.
- S is 100 pages long, and S tuples are 500 bytes long.
- C is a key for S, and A is a key for R.
- The page size is 1024 bytes.
- Each S tuple joins with exactly one R tuple.
- The combined size of attributes A, B, C, and D is 450 bytes.
- A and B are in R and have a combined size of 200 bytes; C and D are in S.

1. What is the cost of writing out the final result? (As usual, you should ignore this cost in answering subsequent questions.)

2. Suppose that three buffer pages are available, the only join method that is implemented is page-oriented nested loops, and 1/10 of the tuples are removed as duplicates after projection.
   (a) Compute the cost of doing the projection followed by the join.
   (b) Compute the cost of doing the join followed by the projection.
   (c) Compute the cost of doing the join first and then the projection on-the-fly.
   (d) Would your answers change if 11 buffer pages were available?

---

\(^1\) Hint: 1. assume tuples within each bucket are uniform; 2. when joining two buckets, each tuple from the smaller one always matches some tuples from the larger one.

\(^2\) $|\text{est}_1 - \text{est}_2|/\text{est}_2 \times 100\%$