Counting inversions

Music site tries to match your song preferences with others.
- You rank n songs.
- Music site consults database to find people with similar tastes.

Similarity metric: number of inversions between two rankings.
- My rank: 1, 2, ..., n.
- Your rank: a_1, a_2, ..., a_n.
- Songs i and j are inverted if i < j, but a_i > a_j.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>me</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>you</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

2 inversions: 3-2, 4-2

Brute force: check all Θ(n^2) pairs.
Divide and conquer approach

- Divide: separate list into two halves $A$ and $B$.
- Conquer: recursively count inversions in each list.
- Combine: count inversions $(a, b)$ with $a \in A$ and $b \in B$.
- Return sum of three counts.

<table>
<thead>
<tr>
<th>Input</th>
<th>Count inversions in left half A</th>
<th>Count inversions in right half B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 5 4 8 10 2 6 9 3 7</td>
<td>5-4</td>
<td>6-3 9-3 9-7</td>
</tr>
<tr>
<td>Count inversions $(a, b)$ with $a \in A$ and $b \in B$</td>
<td>4-2 4-3 5-2 5-3 8-2 8-3 8-6 8-7 10-2 10-3 10-6 10-7 10-9</td>
<td>2 6 9 3 7</td>
</tr>
<tr>
<td>Output</td>
<td>1 + 3 + 13 = 17</td>
<td></td>
</tr>
</tbody>
</table>

How can we do both?

Q. How to count inversions $(a, b)$ with $a \in A$ and $b \in B$?

A. Easy if $A$ and $B$ are sorted!

Warmup algorithm.

- Sort $A$ and $B$.
- For each element $b \in B$,
  - binary search in $A$ to find how elements in $A$ are greater than $b$.

<table>
<thead>
<tr>
<th>List A</th>
<th>List B</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 10 18 3 14</td>
<td>20 23 2 11 16</td>
</tr>
<tr>
<td>Sort A</td>
<td>Sort B</td>
</tr>
<tr>
<td>3 7 10 14 18</td>
<td>2 11 16 20 23</td>
</tr>
<tr>
<td>Binary search to count inversions $(a, b)$ with $a \in A$ and $b \in B$</td>
<td></td>
</tr>
<tr>
<td>3 7 10 14 18</td>
<td>2 11 16 20 23</td>
</tr>
<tr>
<td>5 2 1 0 0</td>
<td></td>
</tr>
</tbody>
</table>
How, contd.

Count inversions \((a, b)\) with \(a \in A\) and \(b \in B\), assuming \(A\) and \(B\) are sorted.
- Scan \(A\) and \(B\) from left to right.
- Compare \(a_i\) and \(b_j\).
- If \(a_i < b_j\), then \(a_i\) is not inverted with any element left in \(B\).
- If \(a_i > b_j\), then \(b_j\) is inverted with every element left in \(A\).
- Append smaller element to sorted list \(C\).

Count inversions \((a, b)\) with \(a \in A\) and \(b \in B\)

\[
\begin{array}{cccc|c}
3 & 7 & 10 & a_i & 15 \\
2 & 11 & b_j & 20 & 23 \\
5 & 2 & & \\
\end{array}
\]

merge to form sorted list \(C\)

\[
\begin{array}{cccc}
2 & 3 & 7 & 10 & 11 \\
\end{array}
\]

Counting conversions - Algorithm

**Input.** List \(L\).

**Output.** Number of inversions in \(L\) and sorted list of elements \(L'\).

\[
\begin{align*}
\text{SORT-AND-COUNT} (L) & \\
\text{IF} \; \text{list} \; L \; \text{has one element} & \\
\text{RETURN} \; (0, L). \\
\text{DIVIDE} \; \text{the list into two halves} \; A \; \text{and} \; B. & \\
(t_A, A) & \leftarrow \text{SORT-AND-COUNT}(A). \\
(t_B, B) & \leftarrow \text{SORT-AND-COUNT}(B). \\
(t_{AB}, L') & \leftarrow \text{MERGE-AND-COUNT}(A, B). \\
\text{RETURN} \; (t_A + t_B + t_{AB}, L'). & \\
\end{align*}
\]