Divide and Conquer: Counting Inversions
Collaborative filtering
- matches your preference (books, music, movies, restaurants) with that of others
- finds people with similar tastes
- recommends new things to you based on purchases of these people

Meta-search tools
- same query to many search engines
- synthesize result by looking for similarities of resulting rankings

The basis: compare the similarity of two rankings
What's similar?

Given numbers 1 to n (the things) rank these according to your preference
- You get some permutation of 1..n
- Compare to someone else's permutation

Extreme similarity
- somebody else's ranking is exactly the same

Extreme dissimilarity
- somebody else's ranking is exactly the opposite

In the middle:
- count the number of out of place rankings
Count the number of **inversions** of a ranking

- \( r_1, r_2, \ldots, r_n \)
- count the number of out of order pairs
  - \( i < j \quad r_i > r_j \)

**eg:** 2 1 4 3 5

- 2 inversions: (2,1) (4,3)

Why is this synonymous with comparing two different rankings?

Because we can re-number, such that one of the rankings becomes \( 1,2,\ldots,n \)
Visualizing inversions

zero inversions

1 2 3 4 5

one inversion

2 1 3 4 5

1 2 3 4 5
Visualizing inversions

how many? 3 2 1 4 5
enumerate them

1 2 3 4 5

how many? 5 2 3 4 1

1 2 3 4 5
Does Bubble sort count inversions?
Selection sort?
Insertion sort?
These are $O(n^2)$

Do these sorts on:
and see what happens
Do it to it

4  2  3  5  1
1  2  3  4  5

2  4  3  5  1
1  2  3  4  5

2  3  4  5  1
1  2  3  4  5

2  3  1  4  5
1  2  3  4  5

2  1  3  4  5
1  2  3  4  5

2  1  3  4  5
1  2  3  4  5
Can we do better?

Notice: there are potentially $n(n-1)/2$ inversions

Bubble and insertion sort count each individual inversion

To do better we must not count each individual inversion

Think of merge sort

- in merge sort we do not swap all elements that are out of order with each other, we make larger distance "swaps"
- if we can merge sort and keep track of the number of inversions we may get an $O(n \log n)$ algorithm
Eg: \[ 4 \ 2 \ 3 \ 5 \ 1 \]

\textbf{sort \ [4 \ 2 \ 3 \ 5 \ 1]} \]

- \textbf{sort LEFT: \ [4 \ 2 \ 3]} \\
  - sort left: \ [4 \ 2] \rightarrow \ [2 \ 4] : 1 \text{ inversion} \\
  - sort right: \ [3] \\
  - merge(left,right) \rightarrow \ [2 \ 3 \ 4] \ 1 \text{ inversions (3 jumps over 4)}

- \textbf{sort RIGHT: \ [5 \ 1]} \rightarrow \ [1 \ 5] \ 1 \text{ inversion}

- \textbf{merge(LEFT,RIGHT) \rightarrow [1 \ 2 \ 3 \ 4 \ 5]} \ 3 \text{ inversions (1 jumps over 2,3 & 4)}

Total inversions: 1+1+1+3=6  (go check the visualization)
The algorithm

While merging in merge sort keep track of the number of inversions. When merging an element from left: no inversions added. When merging an element from right: how many inversions added?

As many elements as are remaining in left, because the element from the right jumps over them.
Counting Inversions: Algorithm

Sort-and-Count(L)

if list L has one element
    return 0 and the list L
divide the list into two halves A and B
    (r_A, A) ← Sort-and-Count(A)
    (r_B, B) ← Sort-and-Count(B)
    (r, R) ← Merge-and-Count(A, B)
return r = r_A + r_B + r and the sorted list R

Merge-and-Count(L,R)

count = 0
while L and R not empty:
    append smallest of Li and Rj to result
    if Rj smallest
        add number of elements remaining in L to count
    if one list empty
        append the other one to result
return count, result
Running time

Just like merge sort, the sort and count algorithm running time satisfies:

\[ T(n) = 2 \ T(n / 2) + cn \]

Running time is therefore \( O(n \log n) \)
Repeated substitution

**Claim.** If \( T(n) \) satisfies this recurrence, then \( T(n) = cn \log_2 n \).

\[
T(n) = \begin{cases} 
  c & \text{if } n = 1 \\
  2T(n/2) + cn & \text{otherwise}
\end{cases}
\]

For \( n > 1 \):

\[
T(n) = 2T(n/2) + cn \\
= 4T(n/4) + cn + 2n/2 \\
= 8T(n/8) + cn + cn + 4cn/4 \\
\vdots \\
= 2^{\log_2 n} T(1) + cn + \cdots + cn \\
= O(n \log_2 n)
\]
mergesort: Recurrence Analysis

\[ f(n) = a \cdot f\left( \frac{n}{b} \right) + cn^d \]

\[
a = \\
b = \\
d = \\
O(?)
\]

\[
f(n) = \begin{cases} 
O(n^d) & \text{if } a < b^d \\
O\left(n^d \log n\right) & \text{if } a = b^d \\
O\left(n^{\log_b a}\right) & \text{if } a > b^d 
\end{cases}
\]
Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

\[ i = 6 \]

\[
\begin{array}{cccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
\end{array}
\]

\[
\begin{array}{cccccc}
2 & 11 & 16 & 17 & 23 & 25 \\
\end{array}
\]

Total: 6
**Merge and Count**

**Merge and count step.**
- Given two sorted halves, count number of inversions where \(a_i\) and \(a_j\) are in different halves.
- Combine two sorted halves into sorted whole.

---

```
i = 6

3 7 10 14 18 19
```

```
2 11 16 17 23 25
```

**two sorted halves**

**auxiliary array**

```
2
```

Total: 6
Merge and Count

Merge and count step.

- Given two sorted halves, count number of inversions where \(a_i\) and \(a_j\) are in different halves.
- Combine two sorted halves into sorted whole.

\[i = 6\]

\[\begin{array}{cccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
\end{array}\]

\[\begin{array}{cccccc}
2 & 11 & 16 & 17 & 23 & 25 \\
\end{array}\]

Total: 6

two sorted halves

auxiliary array
Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

\[ i = 5 \]

\[
\begin{array}{cccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
2 & 11 & 16 & 17 & 23 & 25 \\
6 & & & & & & & \\
\end{array}
\]

Two sorted halves

\[
\begin{array}{cccccc}
2 & 3 & & & & \\
\end{array}
\]

Auxiliary array

Total: 6
Merge and Count

Merge and count step.
- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

\[ i = 5 \]

\[ \begin{array}{cccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
\end{array} \]

\[ \begin{array}{cccccccc}
2 & 11 & 16 & 17 & 23 & 25 \\
\end{array} \]

Total: 6
Merge and count step.
- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

```
i = 4  
3   7   10  14  18  19  
    ↓        ↓
  2   11  16  17  23  25  
    6   
```

- **two sorted halves**
- **auxiliary array**
- **Total: 6**
Merge and Count

Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{array}{cccccc}
3 & 7 & \textbf{10} & 14 & 18 & 19 \\
\downarrow & & & & & & \\
2 & 11 & 16 & 17 & 23 & 25 \\
\end{array}
\]

\[i = 4\]

Total: 6
Merge and count step.
- Given two sorted halves, count number of inversions where \( a_i \) and \( a_j \) are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{array}{ccccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
\downarrow & & & & & & \\
2 & 11 & 16 & 17 & 23 & 25 \\
\end{array}
\]

Total: 6
Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

```
    i = 3

  3  7  10  14  18  19
```

```
  2 11  16  17  23  25
    6    3
```

```
  2  3  7  10  11
```

Total: $6 + 3$
Merge and count step.

- Given two sorted halves, count number of inversions where \( a_i \) and \( a_j \) are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{array}{cccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
\end{array}
\begin{array}{cccccc}
2 & 11 & 16 & 17 & 23 & 25 \\
6 & 3 \\
\end{array}
\]

Total: 6 + 3
Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

```
i = 3
```

```
<table>
<thead>
<tr>
<th>3</th>
<th>7</th>
<th>10</th>
<th>14</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
</table>
```

```
<table>
<thead>
<tr>
<th>2</th>
<th>11</th>
<th>16</th>
<th>17</th>
<th>23</th>
<th>25</th>
</tr>
</thead>
</table>
```

Total: $6 + 3$
Merge and Count

Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

```
<table>
<thead>
<tr>
<th>3</th>
<th>7</th>
<th>10</th>
<th>14</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11</td>
<td>16</td>
<td>17</td>
<td>23</td>
<td>25</td>
</tr>
</tbody>
</table>
```

```
i = 2
```

```
two sorted halves
```

```
2 | 3 | 7 | 10 | 11 | 14 |
```

```
auxiliary array
```

```
Total: 6 + 3
```
Merge and Count

Merge and count step.

- Given two sorted halves, count number of inversions where \( a_i \) and \( a_j \) are in different halves.
- Combine two sorted halves into sorted whole.

\[
i = 2
\]

\[
\begin{array}{ccccccc}
3 & 7 & 10 & 14 & 18 & 19 & \\
\end{array}
\]

\[
\begin{array}{ccccccc}
2 & 11 & 16 & 17 & 23 & 25 & \\
6 & 3 & 2 & \\
\end{array}
\]

Total: \( 6 + 3 + 2 \)
Merge and Count

Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

$$i = 2$$

<table>
<thead>
<tr>
<th>3</th>
<th>7</th>
<th>10</th>
<th>14</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11</td>
<td>16</td>
<td>17</td>
<td>23</td>
<td>25</td>
</tr>
</tbody>
</table>

Total: $6 + 3 + 2$
**Merge and Count**

**Merge and count step.**
- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

<table>
<thead>
<tr>
<th>3</th>
<th>7</th>
<th>10</th>
<th>14</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>11</th>
<th>16</th>
<th>17</th>
<th>23</th>
<th>25</th>
</tr>
</thead>
</table>

$i = 2$

- **Total:** $6 + 3 + 2 + 2$
Merge and Count

Merge and count step.

- Given two sorted halves, count number of inversions where \( a_i \) and \( a_j \) are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{array}{cccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
\end{array}
\quad i = 2
\quad \begin{array}{cccccc}
2 & 11 & 16 & 17 & 23 & 25 \\
\end{array}
\]

```
2 3 7 10 11 14 16 17
```

Total: \( 6 + 3 + 2 + 2 \)
Merge and Count

Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{array}{ccccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
6 & 2 & 11 & 16 & 17 & 23 & 25 \\
\end{array}
\]

\[
i = 2
\]

Total: $6 + 3 + 2 + 2$
Merge and Count

Merge and count step.

- Given two sorted halves, count number of inversions where \( a_i \) and \( a_j \) are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{align*}
&\text{two sorted halves} \\
&i = 1 \\
&\downarrow \\
&3 \quad 7 \quad 10 \quad 14 \quad 18 \quad 19 \\
&\downarrow \\
&2 \quad 11 \quad 16 \quad 17 \quad 23 \quad 25 \\
&6 \quad 3 \quad 2 \quad 2 \\
\end{align*}
\]

\[
\begin{align*}
&\text{auxiliary array} \\
&2 \quad 3 \quad 7 \quad 10 \quad 11 \quad 14 \quad 16 \quad 17 \quad 18 \\
&\text{Total: } 6 + 3 + 2 + 2
\end{align*}
\]
Merge and count step.

- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

Total: $6 + 3 + 2 + 2$
Merge and Count

Merge and count step.
- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{array}{cccccccc}
3 & 7 & 10 & 14 & 18 & 19 & 2 & 11 & 16 & 17 & 23 & 25 \\
6 & 3 & 2 & 2 & & & & & & & &
\end{array}
\]

First half exhausted $i = 0$

\[
\begin{array}{cccccccc}
2 & 3 & 7 & 10 & 11 & 14 & 16 & 17 & 18 & 19 & \quad \text{auxiliary array}
\end{array}
\]

Total: $6 + 3 + 2 + 2$
Merge and count step.

- Given two sorted halves, count number of inversions where \( a_i \) and \( a_j \) are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{array}{cccccc}
3 & 7 & 10 & 14 & 18 & 19 \\
\end{array}
\quad
\begin{array}{cccccc}
2 & 11 & 16 & 17 & 23 & 25 \\
6 & 3 & 2 & 2 & 0 & \\
\end{array}
\quad
\begin{array}{cccccc}
2 & 3 & 7 & 10 & 11 & 14 & 16 & 17 & 18 & 19 & 23 \\
\end{array}
\quad
\text{two sorted halves}
\quad
\text{auxiliary array}
\quad
\text{Total: } 6 + 3 + 2 + 2 + 0
\]
Merge and count step.

- Given two sorted halves, count number of inversions where \( a_i \) and \( a_j \) are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{align*}
\text{i} &= 0 \\
3 &\quad 7 &\quad 10 &\quad 14 &\quad 18 &\quad 19 \\
2 &\quad 11 &\quad 16 &\quad 17 &\quad 23 &\quad 25 \\
6 &\quad 3 &\quad 2 &\quad 2 &\quad 0
\end{align*}
\]

\[
\begin{align*}
2 &\quad 3 &\quad 7 &\quad 10 &\quad 11 &\quad 14 &\quad 16 &\quad 17 &\quad 18 &\quad 19 &\quad 23 \\
\text{Total: } 6 + 3 + 2 + 2 + 0
\end{align*}
\]
Merge and Count

Merge and count step.
- Given two sorted halves, count number of inversions where $a_i$ and $a_j$ are in different halves.
- Combine two sorted halves into sorted whole.

$i = 0$

<table>
<thead>
<tr>
<th>3</th>
<th>7</th>
<th>10</th>
<th>14</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>11</th>
<th>16</th>
<th>17</th>
<th>23</th>
<th>25</th>
</tr>
</thead>
</table>

Total: $6 + 3 + 2 + 2 + 0 + 0$

Two sorted halves

Auxiliary array
Merge and count step.

- Given two sorted halves, count number of inversions where \( a_i \) and \( a_j \) are in different halves.
- Combine two sorted halves into sorted whole.

\[
\begin{array}{cccccccc}
3 & 7 & 10 & 14 & 18 & 19 & & \\
2 & 11 & 16 & 17 & 23 & 25 & & \\
6 & 3 & 2 & 2 & 0 & 0 & & \\
\end{array}
\]

Total: \( 6 + 3 + 2 + 2 + 0 + 0 = 13 \)