Counting Inversions

Web site tries to match your preferences with others on Internet.

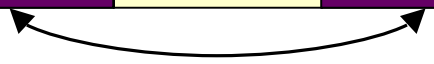
- You rank N songs.
- Web site consults database to find people with similar rankings.

Closeness metric.

- My rank = { 1, 2, . . . , N }.
- Your rank = { a_1, a_2, . . . , a_N }.
- Number of inversions between two preference lists.
- Songs i and j inverted if i < j, but $a_i > a_j$

Songs

	A	B	C	D	E
Me	1	2	3	4	5
You	1	3	4	2	5



Inversion

Inversions

{3, 2}, {4, 2}

Counting Inversions

Brute-force solution.

- **Check all pairs i and j such that i < j.**
- **Θ (N^2) comparisons.**

Note: there can be a quadratic number of inversions.

- **Asymptotically faster algorithm must compute total number without even looking at each inversion individually.**

Counting Inversions: Divide-and-Conquer

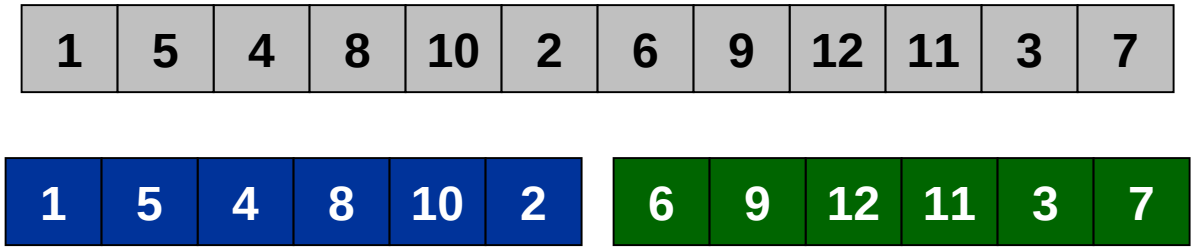
Divide-and-conquer.

1	5	4	8	10	2	6	9	12	11	3	7

Counting Inversions: Divide-and-Conquer

Divide-and-conquer.

- **Divide: separate list into two pieces.**

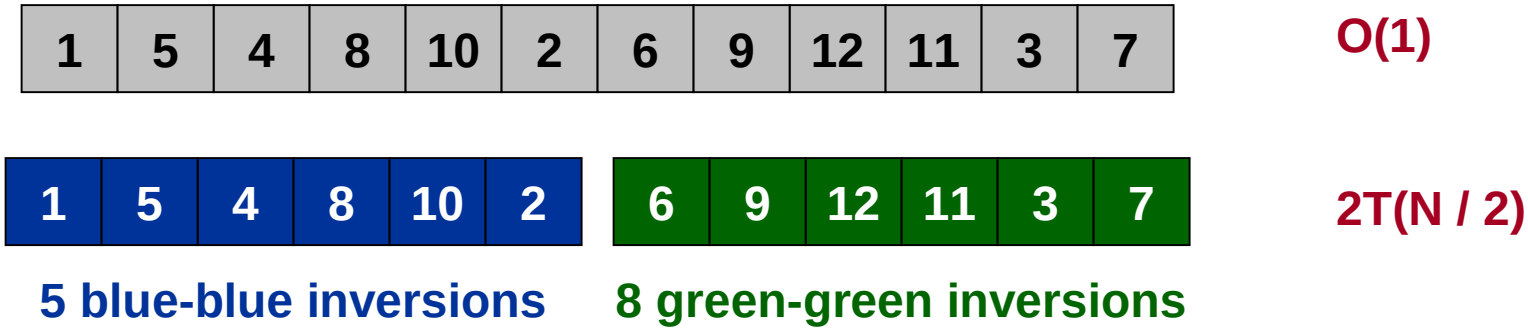


O(1)

Counting Inversions: Divide-and-Conquer

Divide-and-conquer.

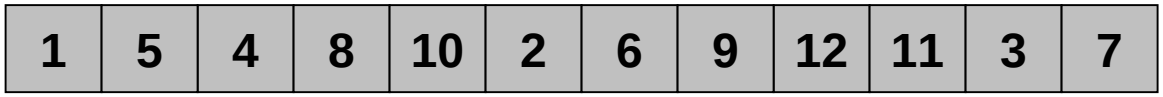
- **Divide: separate list into two pieces.**
- **Conquer: recursively count inversions in each half separately.**



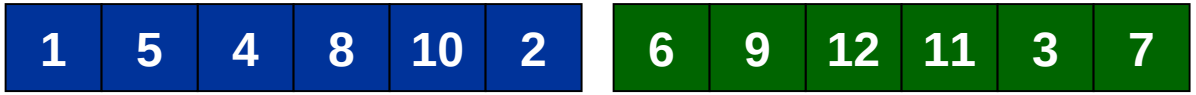
Counting Inversions: Divide-and-Conquer

Divide-and-conquer.

- **Divide: separate list into two pieces.**
- **Conquer: recursively count inversions in each half.**
- **Combine: count inversions where a_i and a_j are in different halves.**



O(1)



2T(N / 2)

5 blue-blue inversions **8 green-green inversions**

9 blue-green inversions:
{5-3, 4-3, 8-6, 8-3, 8-7, 10-6, 10-9, 10-3, 10-7}

$O(N^2)$

Counting Inversions: Divide-and-Conquer

Divide-and-conquer.

- **Divide: separate list into two pieces.**
- **Conquer: recursively count inversions in each half.**
- **Combine: count inversions where a_i and a_j are in different halves.**
- **Return sum of three quantities.**

1	5	4	8	10	2	6	9	12	11	3	7

O(1)

1	5	4	8	10	2

6	9	12	11	3	7

2T(N / 2)

5 blue-blue inversions **8 green-green inversions**

9 blue-green inversions:
{5-3, 4-3, 8-6, 8-3, 8-7, 10-6, 10-9, 10-3, 10-7}

$O(N^2)$

Total = 5 + 8 + 9 = 22.

O(1)

Counting Inversions: Divide-and-Conquer

Divide-and-conquer.

- **Divide: separate list into two pieces.**
- **Conquer: recursively count inversions in each half.**
- **Combine: count inversions where a_i and a_j are in different halves.**
- **Return sum of three quantities.**

1	5	4	8	10	2	6	9	12	11	3	7

O(1)

1	5	4	8	10	2

6	9	12	11	3	7

2T(N / 2)

5 blue-blue inversions **8 green-green inversions**

9 blue-green inversions:
{5-3, 4-3, 8-6, 8-3, 8-7, 10-6, 10-9, 10-3, 10-7}

Can we do this step in sub-quadratic time?

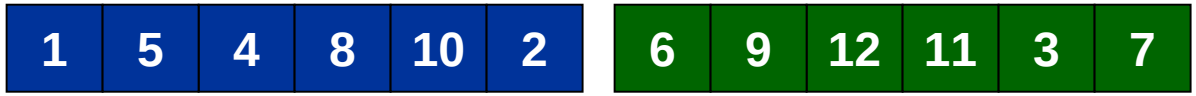
Total = 5 + 8 + 9 = 22.

O(1)

Counting Inversions: Good Combine

Combine: count inversions where a_i and a_j are in different halves.

- **Key idea: easy if each half is sorted.**
- **Sort each half.**
- **Count inversions.**



O(N log N)

9 blue-green inversions: 4 + 2 + 2 + 1 + 0 + 0

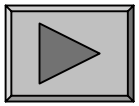
O(N)

$$T(N) = T(\lfloor N/2 \rfloor) + T(\lceil N/2 \rceil) + O(N \log N) \Rightarrow T(N) = O(N \log^2 N)$$

Counting Inversions: Better Combine

Combine: count inversions where a_i and a_j are in different halves.

- **Assume each half is pre-sorted.**
- **Count inversions.**
- **Merge two sorted halves into sorted whole.**



3	7	10	14	18	19

2	11	16	17	23	25

13 blue-green inversions: 6 + 3 + 2 + 2 + 0 + 0

O(N)

2	3	7	10	11	14	16	17	18	19	23	25

O(N)

$$T(N) = T(\lfloor N/2 \rfloor) + T(\lceil N/2 \rceil) + O(N) \Rightarrow T(N) = O(N \log N)$$