Plan for today

quiz

discuss quiz

two more divide and conquer algorithms

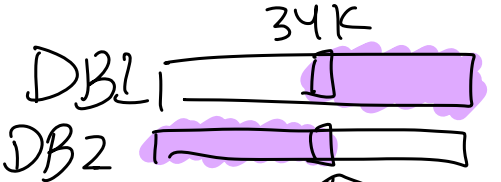
+ activity

end at

9:50

(a) If database 1 holds [\$10000, \$18000, \$25000, \$80000] and database 2 holds [\$30000, \$38000, \$45000, \$75000], what is the median household income? (You can report either the n^{th} or the $n+1^{\text{th}}$ value in the combined list.)

10k 18k 25k 30k 38k 45k 75k 80k



(b) Now consider some new, large-size instance of the problem, with an even n. Suppose that you query for the $n/2^{\text{th}}$ value in database 1 and find that it is \$34000, and you query for the $n/2^{\text{th}}$ value in database 2 and find that it is \$46000. What do you know about the median of the combined lists? Circle one:

46k

- The median of the combined lists is in the first half of database 1 and the first half of database 2.
- The median of the combined lists is in the first half of database 1 and the second half of database 2.
- The median of the combined lists is in the second half of database 1 and the first half of database 2.
- The median of the combined lists is in the second half of database 1 and the second half of database 2.

Overall Median (DB1, DB2, n):

m_1 = query(DB1, $n/2$)

m_2 = query(DB2, $n/2$)

if $m_1 > m_2$:

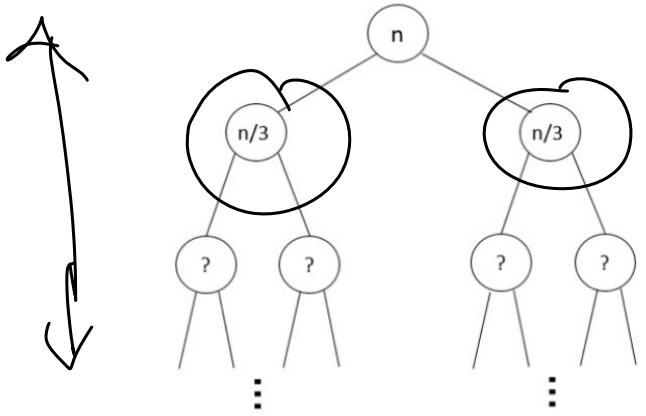
Overall Median(half of DB1, half of DB2, $n/2$)

else:

Overall Median(half of DB1, half of DB2, $n/2$)

What is $T(n)$, # queries? $T(n) = T(n/2) + 2$

2. (8 points) Consider the recursion tree below, which represents the recursive calls (each node) and size of the input to the recursive calls (text inside each node).



Level 0 — $1(n)$

Level 1 — $2(n/3)$

Level 2 — $4(n/3^2) = 4(\frac{n}{4})$

$T(n) = aT(n/b) + f(n) = 2T(n/3) + n$

$a = 2$

$b = 3$

depth? $\log_3 n$

$(\log_3 n)(\quad)$

if $f(n) = n$, # comps at L0, L1, L2, L3 ?

Counting inversions

Counting inversions

Music site tries to match your song preferences with others.

Counting inversions

Music site tries to match your song preferences with others.

- You rank n songs.

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.

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.

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: $1, 2, \ldots, n$.

	A	B	C	D	E
me	1	2	3	4	5

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: $1, 2, \ldots, n$.
- Your rank: $a_1, a_2, \ldots, a_n$.

	A	B	C	D	E
me	1	2	3	4	5
you	1	3	4	2	5

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: $1, 2, \ldots, n$.
- Your rank: $a_1, a_2, \ldots, a_n$.
- Songs i and j are inverted if $i < j$, but $a_i > a_j$.

	A	B	C	D	E
me	1	2	3	4	5
you	1	3	4	2	5

2 inversions: 3-2, 4-2

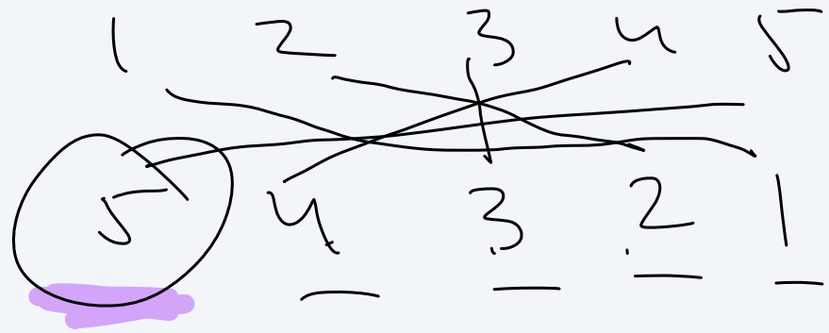
Counting inversions

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: $1, 2, \ldots, n$.
- Your rank: $a_1, a_2, \ldots, a_n$.
- Songs i and j are inverted if $i < j$, but $a_i > a_j$.



With your table:

In a list of length n, what is the minimum number of inversions? How about the maximum?

Give a $O(n^2)$ algorithm to count the number of inversions in a length n list.

1 2 3 4 5 8

5, 8, 3, 2, 1, 4

Counting inversions: divide-and-conquer

n log n

- Divide: separate list into two halves A and B.

input

1	5	4	8	10	2	6	9	3	7

Counting inversions: divide-and-conquer

- Divide: separate list into two halves A and B.
- Conquer: recursively count inversions in each list.

input

1	5	4	8	10	2	6	9	3	7

count inversions in left half A

1	5	4	8	10

count inversions in right half B

2	6	9	3	7

Counting inversions: divide-and-conquer

- 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 $b \in B$.

input

1	5	4	8	10	2	6	9	3	7

count inversions in left half A

1	5	4	8	10

count inversions in right half B

2	6	9	3	7

count inversions (a, b) with a ∈ A and b ∈ B

1	5	4	8	10

2	6	9	3	7

Counting inversions: divide-and-conquer

- 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.

input

1	5	4	8	10	2	6	9	3	7

count inversions in left half A

1	5	4	8	10

count inversions in right half B

2	6	9	3	7

count inversions (a, b) with a ∈ A and b ∈ B

1	5	4	8	10

2	6	9	3	7

Counting inversions: divide-and-conquer

- 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.

input

1	5	4	8	10	2	6	9	3	7

count inversions in left half A

1	5	4	8	10

5-4

count inversions in right half B

2	6	9	3	7

6-3 9-3 9-7

count inversions (a, b) with a ∈ A and b ∈ B

1	5	4	8	10

2	6	9	3	7

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

output 1 + 3 + 13 = 17

Counting inversions: how to combine two subproblems?

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

list A

7	10	18	3	14

list B

20	23	2	11	16

Counting inversions: how to combine two subproblems?

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

A. Easy if A and B are sorted!

list A

7	10	18	3	14

list B

20	23	2	11	16

Counting inversions: how to combine two subproblems?

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.

list A

7	10	18	3	14

list B

20	23	2	11	16

Counting inversions: how to combine two subproblems?

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.

list A

7	10	18	3	14

list B

20	23	2	11	16

sort A

3	7	10	14	18

sort B

2	11	16	20	23

Counting inversions: how to combine two subproblems?

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.

list A

7	10	18	3	14

list B

20	23	2	11	16

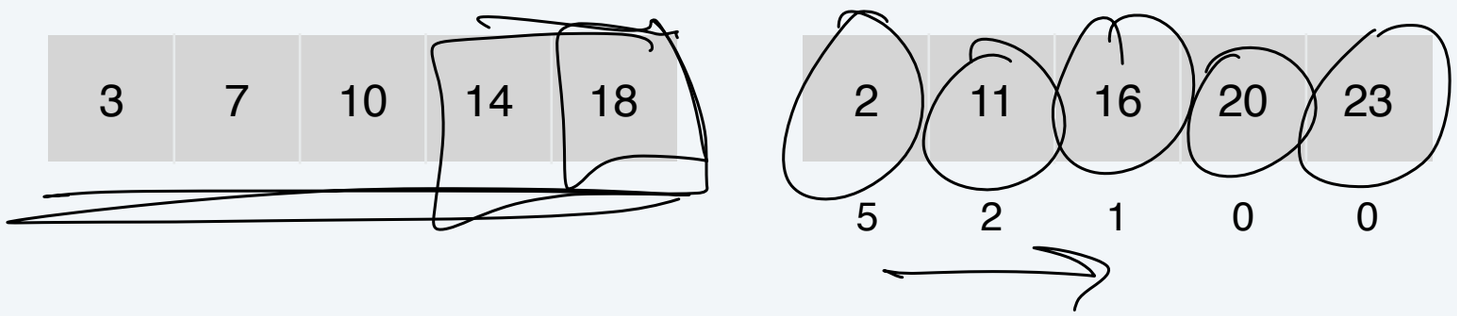
sort A

3	7	10	14	18

sort B

2	11	16	20	23

binary search to count inversions (a, b) with a ∈ A and b ∈ B



Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

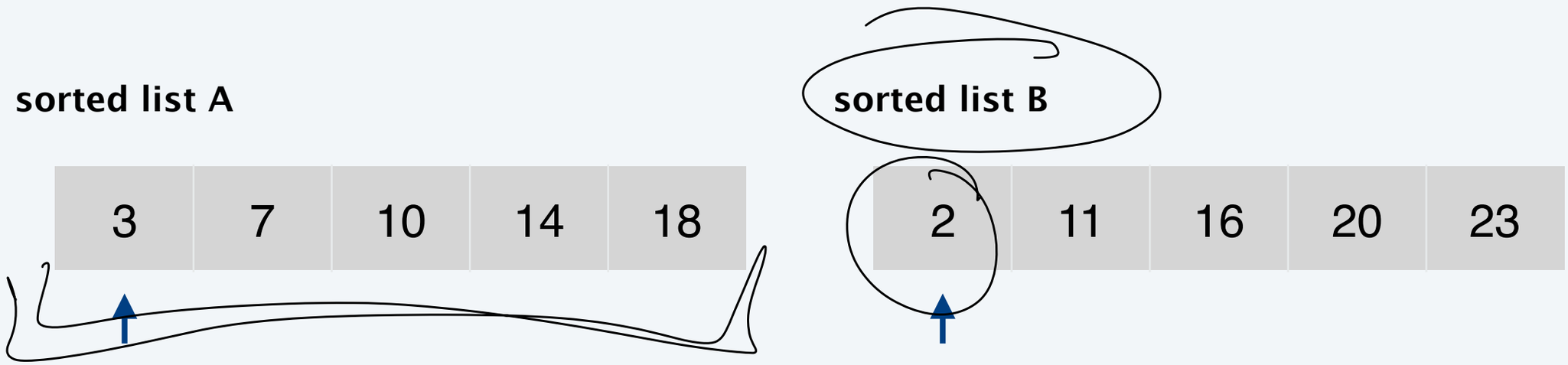
sorted list B

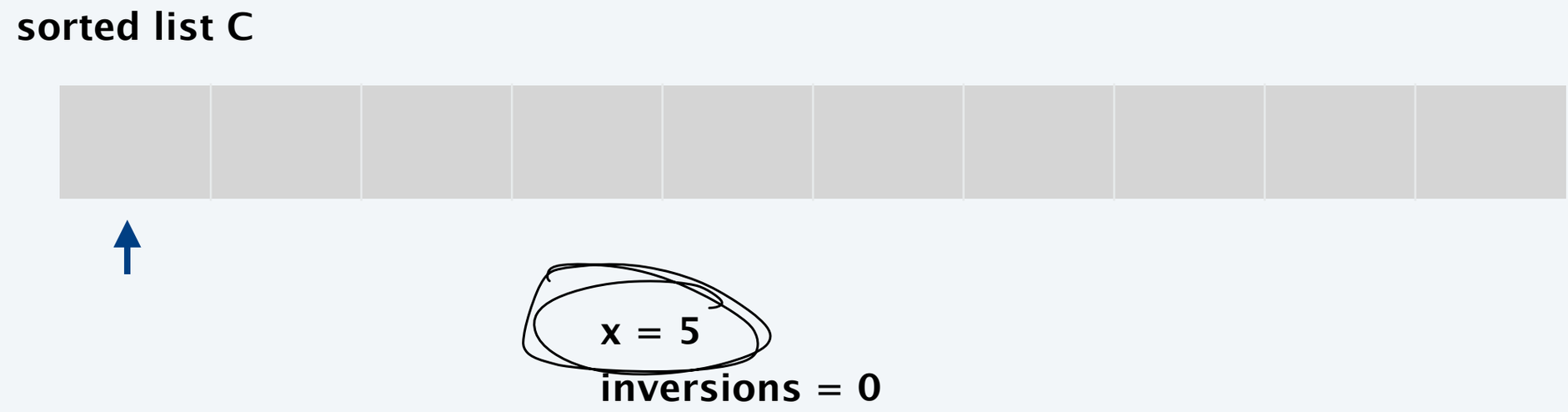
2	11	16	20	23

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.





Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

compare minimum entry in each list: copy 2 and add x to inversion count

sorted list C

x = 5

inversions = 0

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

5	11	16	20	23

sorted list C

2									

x = 5

inversions = 5

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5

compare minimum entry in each list: copy 3 and decrement x

sorted list C

2									

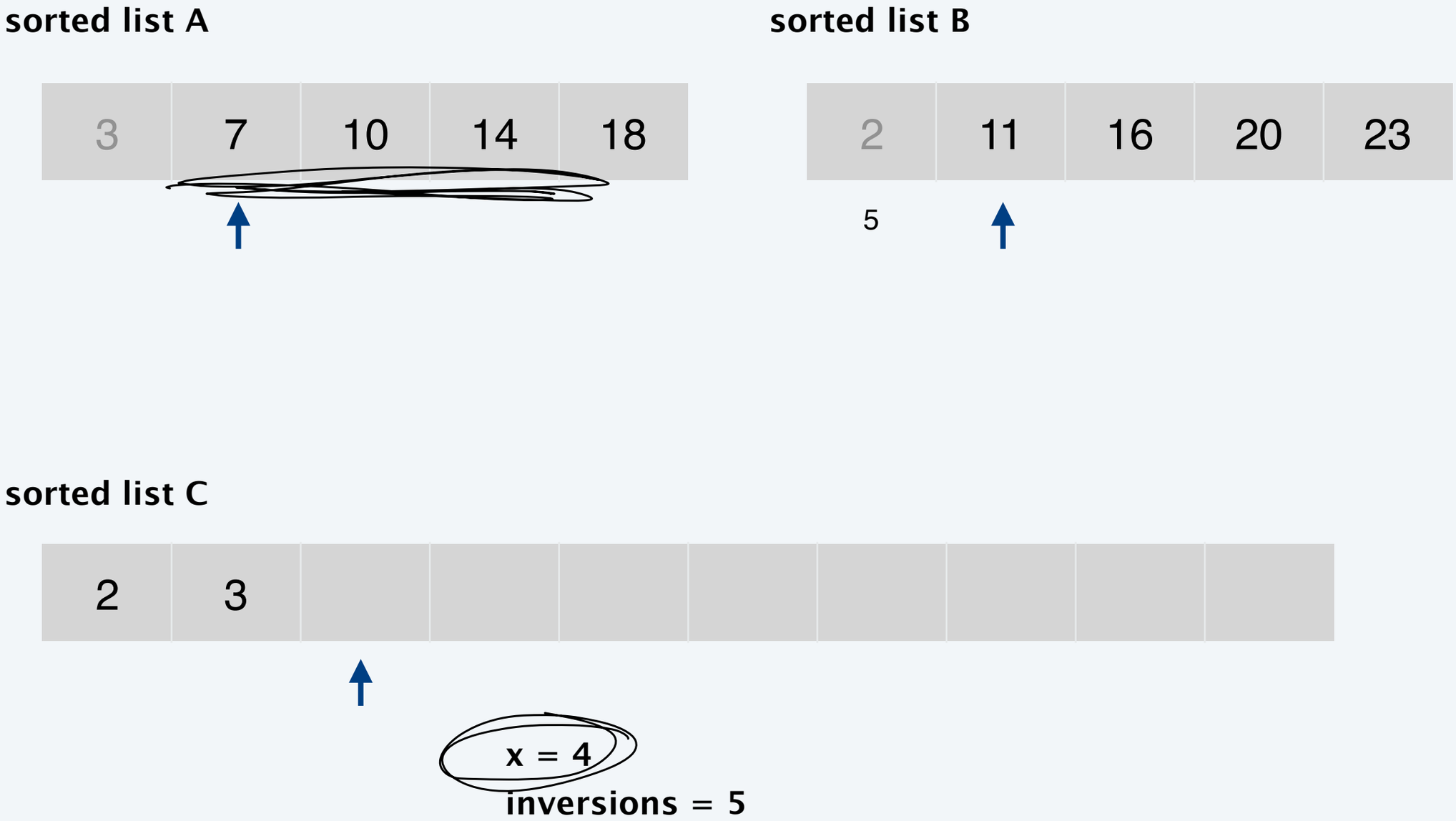
x = 5

inversions = 5

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.



Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5

compare minimum entry in each list: copy 7 and decrement x

sorted list C

2	3								

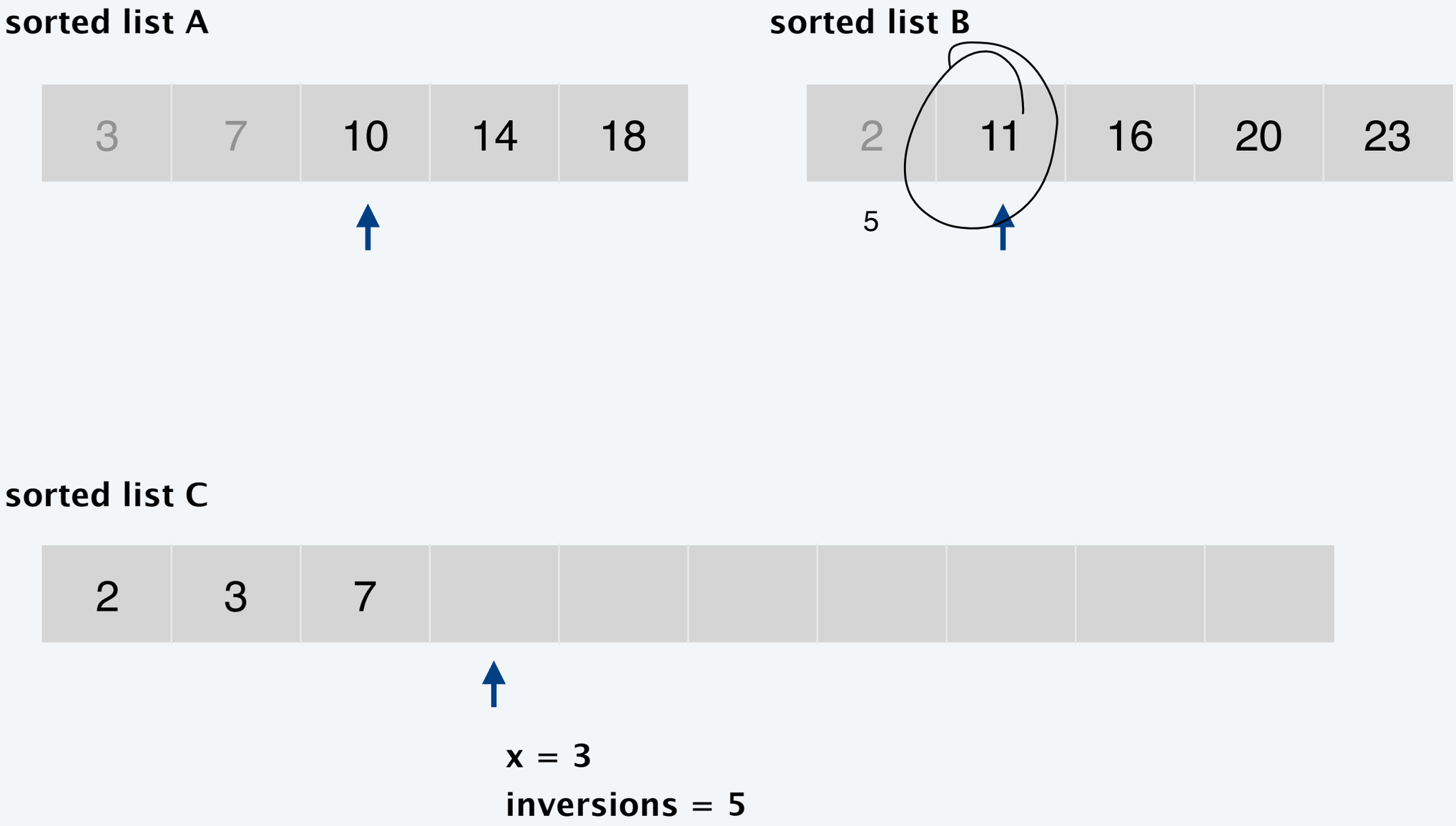
x = 4

inversions = 5

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.



Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5

compare minimum entry in each list: copy 10 and decrement x

sorted list C

2	3	7							

x = 3

inversions = 5

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5

sorted list C

2	3	7	10						

x = 2

inversions = 5

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5

compare minimum entry in each list: copy 11 and add x to increment count

sorted list C

2	3	7	10						

x = 2

inversions = 5

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2

sorted list C

2	3	7	10	11					

x = 2

inversions = 7

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2

compare minimum entry in each list: copy 14 and decrement x

sorted list C

2	3	7	10	11					

x = 2

inversions = 7

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2

sorted list C

2	3	7	10	11	14				

x = 1

inversions = 7

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2

compare minimum entry in each list: copy 16 and add x to increment count

sorted list C

2	3	7	10	11	14				

x = 1

inversions = 7

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2 1

sorted list C

2	3	7	10	11	14	16			

x = 1

inversions = 8

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2 1

compare minimum entry in each list: copy 18 and decrement x

sorted list C

2	3	7	10	11	14	16			

x = 1

inversions = 8

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2 1

sorted list C

2	3	7	10	11	14	16	18		

x = 0

inversions = 8

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2 1

list A exhausted: copy 20

sorted list C

2	3	7	10	11	14	16	18		

x = 0

inversions = 8

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2 1 0

sorted list C

2	3	7	10	11	14	16	20	18	

x = 0

inversions = 8

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2 1 0

list A exhausted: copy 23

sorted list C

2	3	7	10	11	14	16	20	18	

x = 0

inversions = 8

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2 1 0 0

sorted list C

2	3	7	10	11	14	16	20	18	23

x = 0

inversions = 8

Merge and count demo

Given two sorted lists A and B,

- Count number of inversions (a, b) with $a \in A$ and $b \in B$.
- Merge A and B into sorted list C.

sorted list A

3	7	10	14	18

sorted list B

2	11	16	20	23

5 2 1 0 0

done: return 8 inversions

sorted list C

2	3	7	10	11	14	16	20	18	23

x = 0

inversions = 8

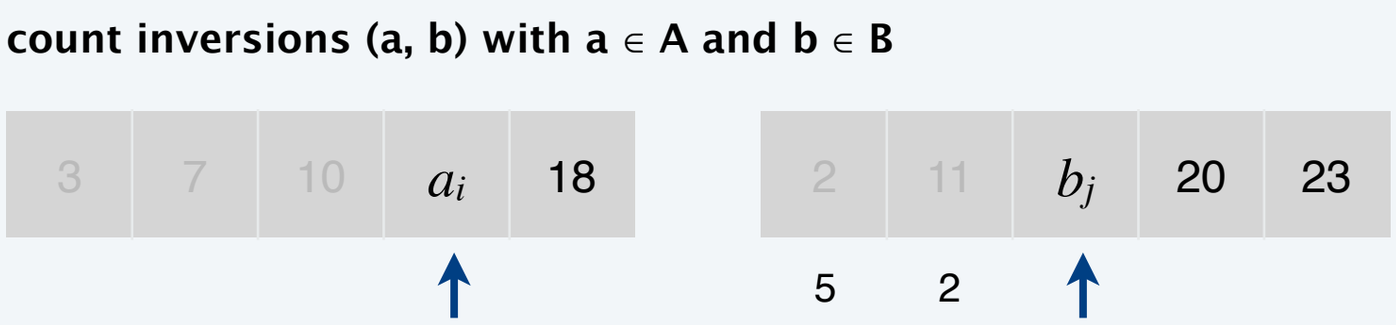
Counting inversions: how to combine two subproblems?

Count inversions (a, b) with $a \in A$ and $b \in B$, assuming A and B are sorted.

Counting inversions: how to combine two subproblems?

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.



Counting inversions: divide-and-conquer algorithm implementation

Input. List L.

Output. Number of inversions in L and L in sorted order.

Counting inversions: divide-and-conquer algorithm implementation

Input. List L.

Output. Number of inversions in L and L in sorted order.

SORT-AND-COUNT(L)

Counting inversions: divide-and-conquer algorithm implementation

Input. List L.

Output. Number of inversions in L and L in sorted order.

SORT-AND-COUNT(L)

IF (list L has one element)

RETURN $(0, L)$.

Counting inversions: divide-and-conquer algorithm implementation

Input. List L.

Output. Number of inversions in L and L in sorted order.

SORT-AND-COUNT(L)

IF (list L has one element)

RETURN $(0, L)$.

Divide the list into two halves A and B.

Counting inversions: divide-and-conquer algorithm implementation

Input. List L.

Output. Number of inversions in L and L in sorted order.

SORT-AND-COUNT(L)

IF (list L has one element)

  RETURN $(0, L)$.

Divide the list into two halves A and B.

$(r_A,\ A) \leftarrow$ SORT-AND-COUNT(A). ⟵ $T(n / 2)$

Counting inversions: divide-and-conquer algorithm implementation

Input. List L.

Output. Number of inversions in L and L in sorted order.

SORT-AND-COUNT(L)

IF (list L has one element)

  RETURN $(0, L)$.

Divide the list into two halves A and B.

$(r_A,\ A) \leftarrow$ SORT-AND-COUNT(A). ⟵ $T(n / 2)$

$(r_B,\ B) \leftarrow$ SORT-AND-COUNT(B). ⟵ $T(n / 2)$

Counting inversions: divide-and-conquer algorithm implementation

Input. List L.

Output. Number of inversions in L and L in sorted order.

SORT-AND-COUNT(L)

IF (list L has one element)

  RETURN $(0, L)$.

Divide the list into two halves A and B.

$(r_A,\ A) \leftarrow$ SORT-AND-COUNT(A). ⟵ $T(n / 2)$

$(r_B,\ B) \leftarrow$ SORT-AND-COUNT(B). ⟵ $T(n / 2)$

$(r_{AB},\ L) \leftarrow$ MERGE-AND-COUNT(A, B). ⟵ $\Theta(n)$

Counting inversions: divide-and-conquer algorithm implementation

Input. List L.

Output. Number of inversions in L and L in sorted order.

SORT-AND-COUNT(L)

IF (list L has one element)

  RETURN $(0, L)$.

Divide the list into two halves A and B.

$(r_A,\ A) \leftarrow$ SORT-AND-COUNT(A). ⟵ $T(n / 2)$

$(r_B,\ B) \leftarrow$ SORT-AND-COUNT(B). ⟵ $T(n / 2)$

$(r_{AB},\ L) \leftarrow$ MERGE-AND-COUNT(A, B). ⟵ $\Theta(n)$

RETURN $(r_A + r_B + r_{AB},\ L)$.

Counting inversions: divide-and-conquer algorithm analysis

Proposition. The sort-and-count algorithm counts the number of inversions in a permutation of size n in $O(n \log n)$ time.

Counting inversions: divide-and-conquer algorithm analysis

Proposition. The sort-and-count algorithm counts the number of inversions in a permutation of size n in $O(n \log n)$ time.

Pf. The worst-case running time $T(n)$ satisfies the recurrence:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n) & \text{if } n > 1 \end{cases}$$

Activity