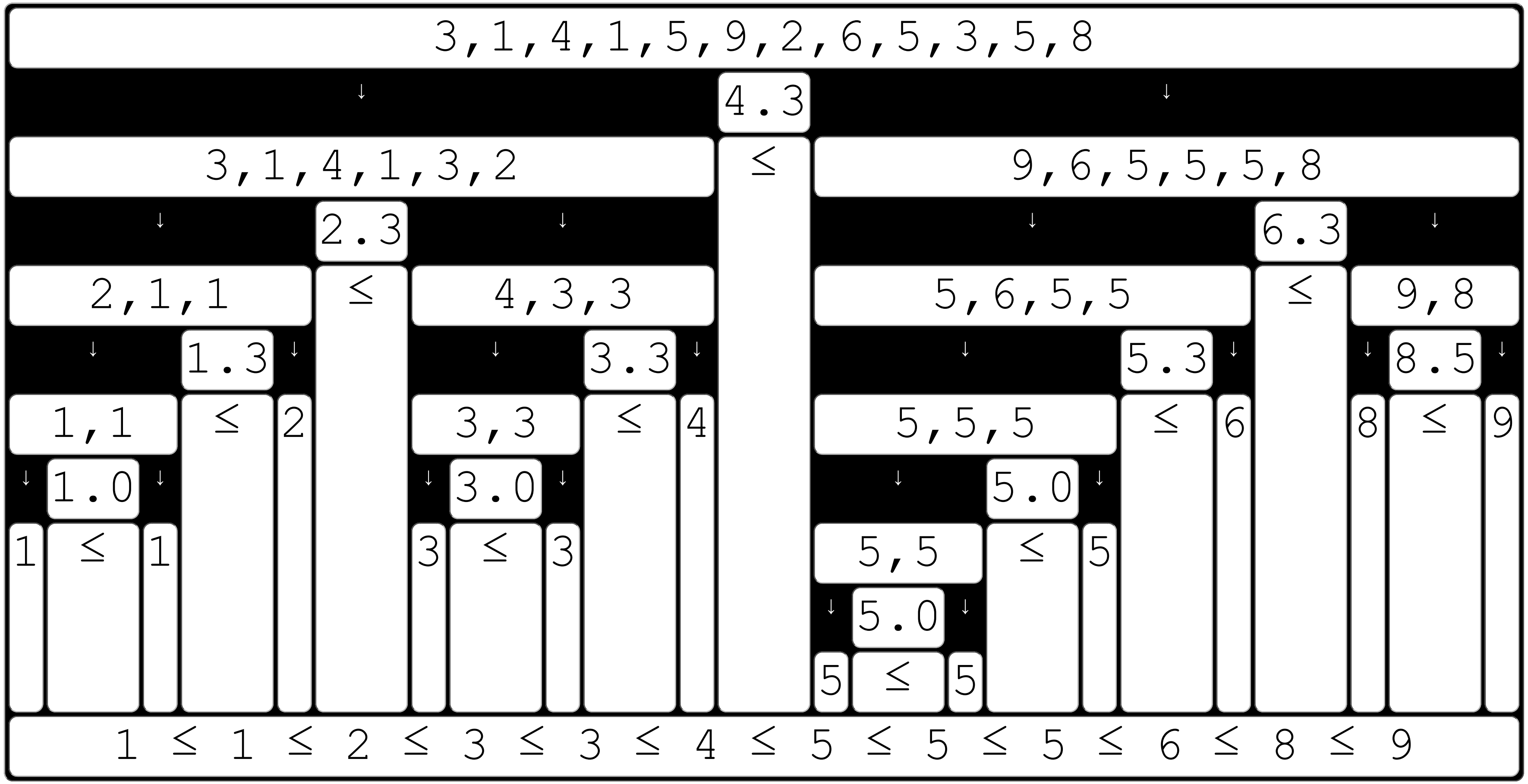
Quicksort
Tutorial Highlight

Julio M. Otuyama

otuyama@alumni.usp.br



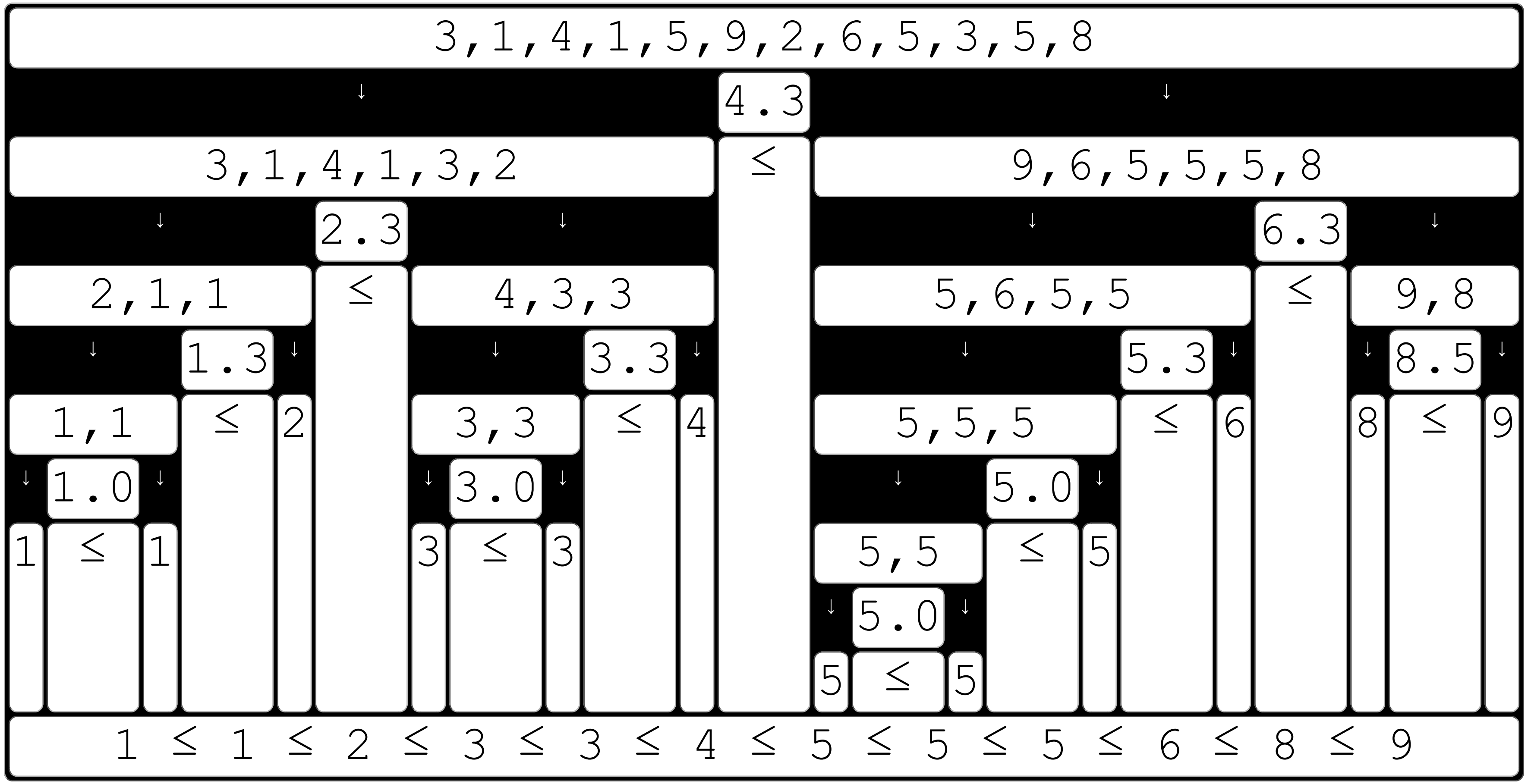
previous

first

Welcome to Quicksort Interactive Tutorial.

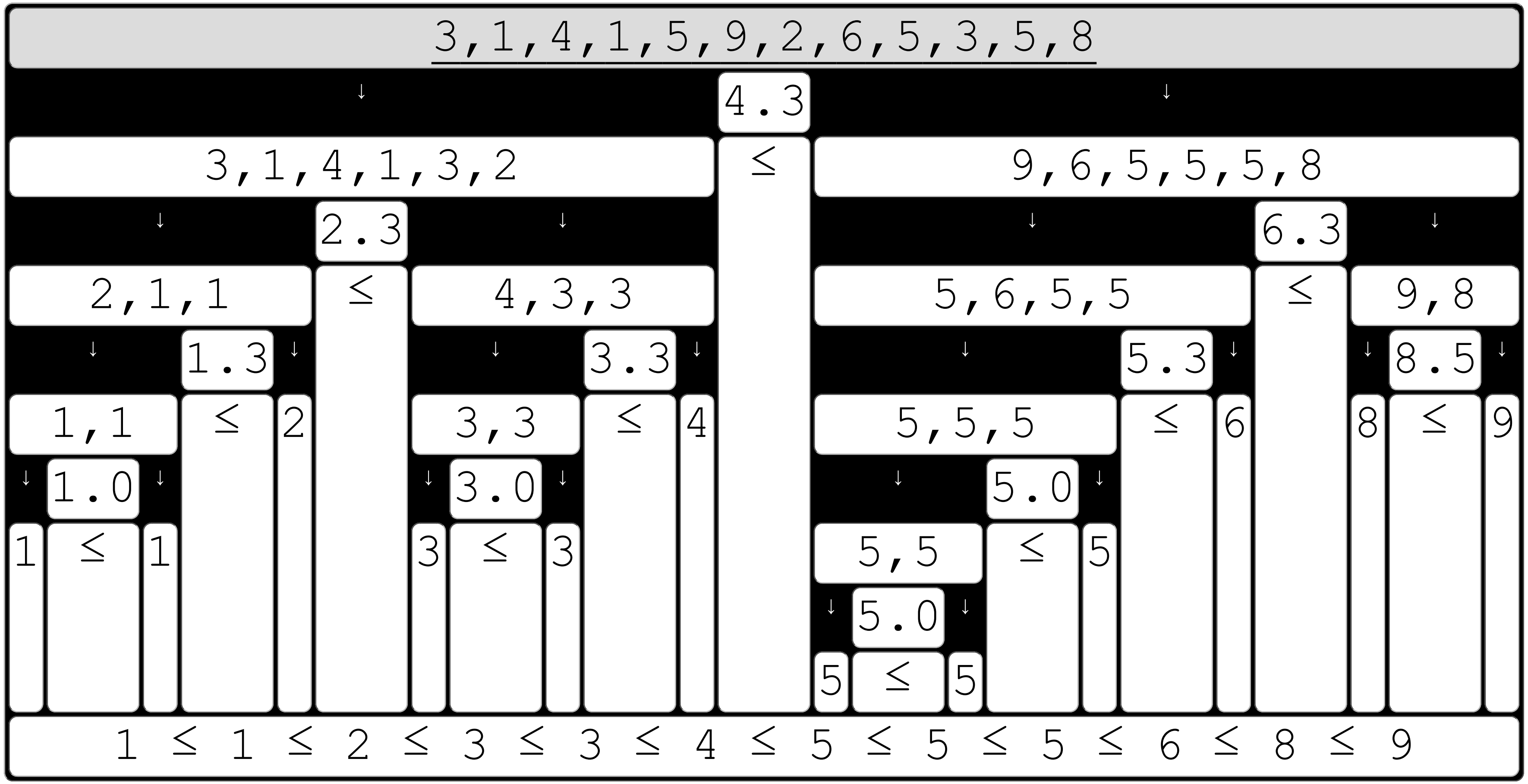
Input your own values: 3,1,4,1,5,9,2,6,5,3,5,8 Sort

next



(1) Introduction.

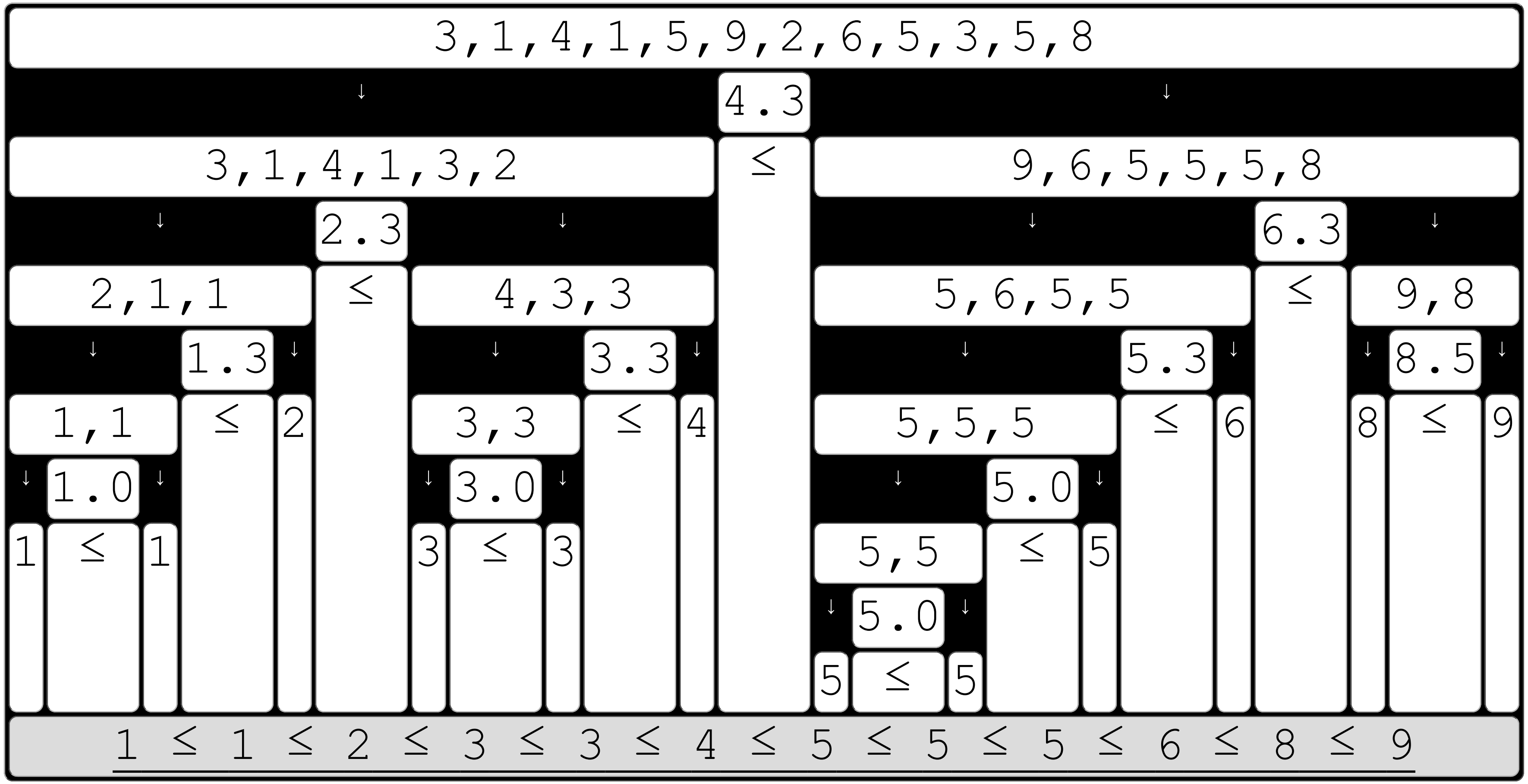
previous
first
next



previous
first

At the top of the diagram, an initial list contains random values (not sorted).

next

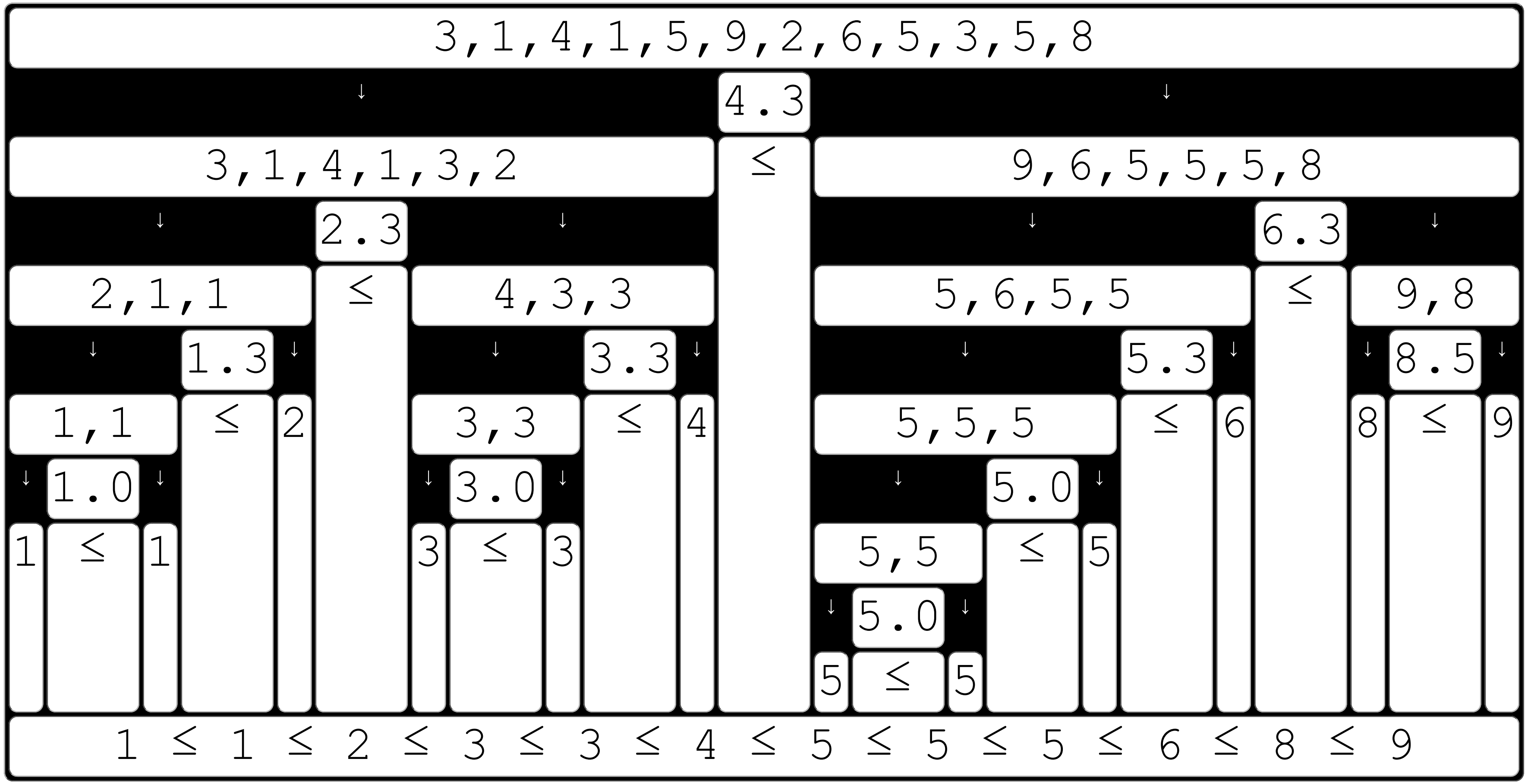


previous

first

At the bottom of the diagram, a sorted list is the result of the algorithm.

next

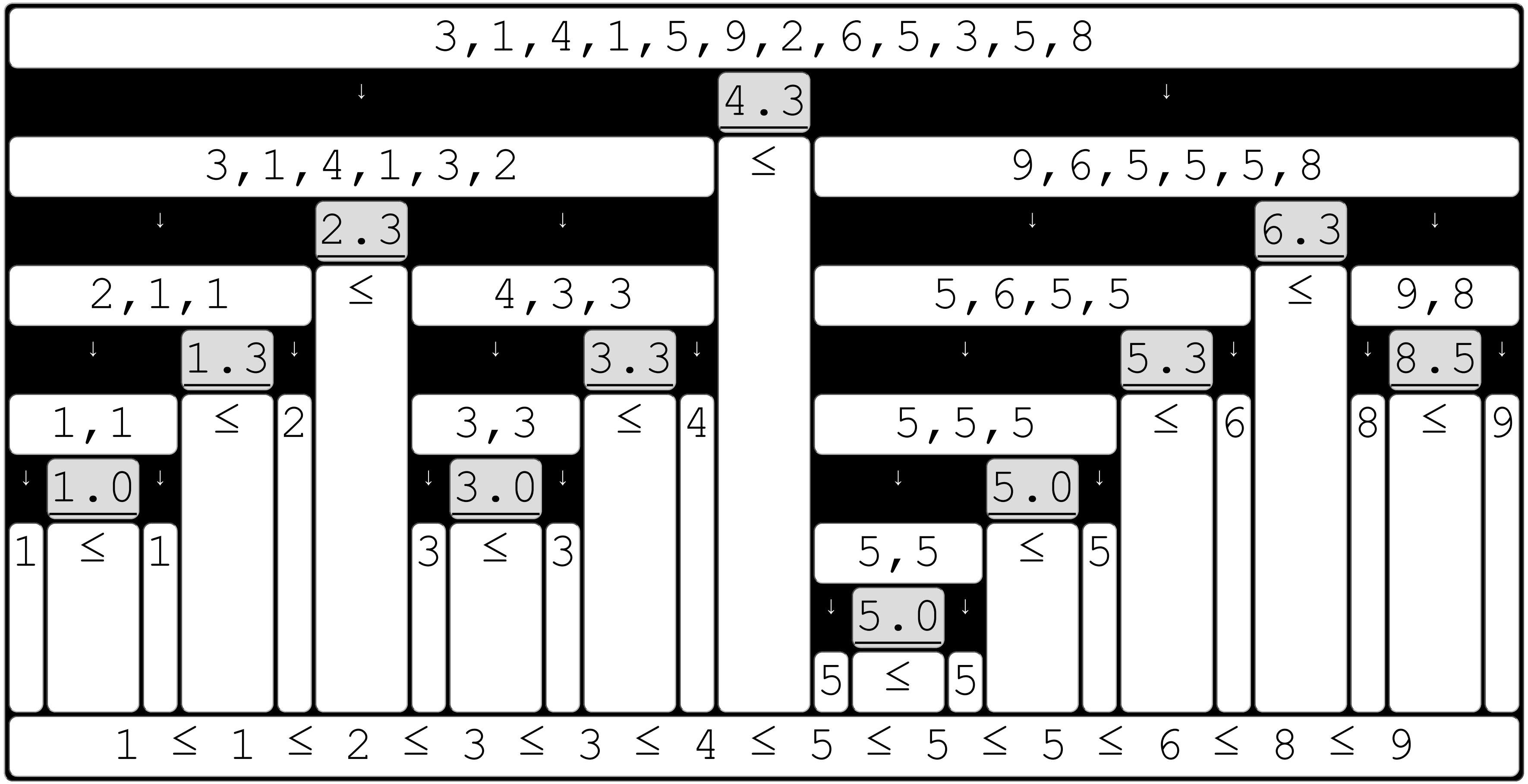


previous

first

Quicksort uses the concept of pivots to split each list near the middle.

next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

↓ 5.0 ↓

5 ≤ 5

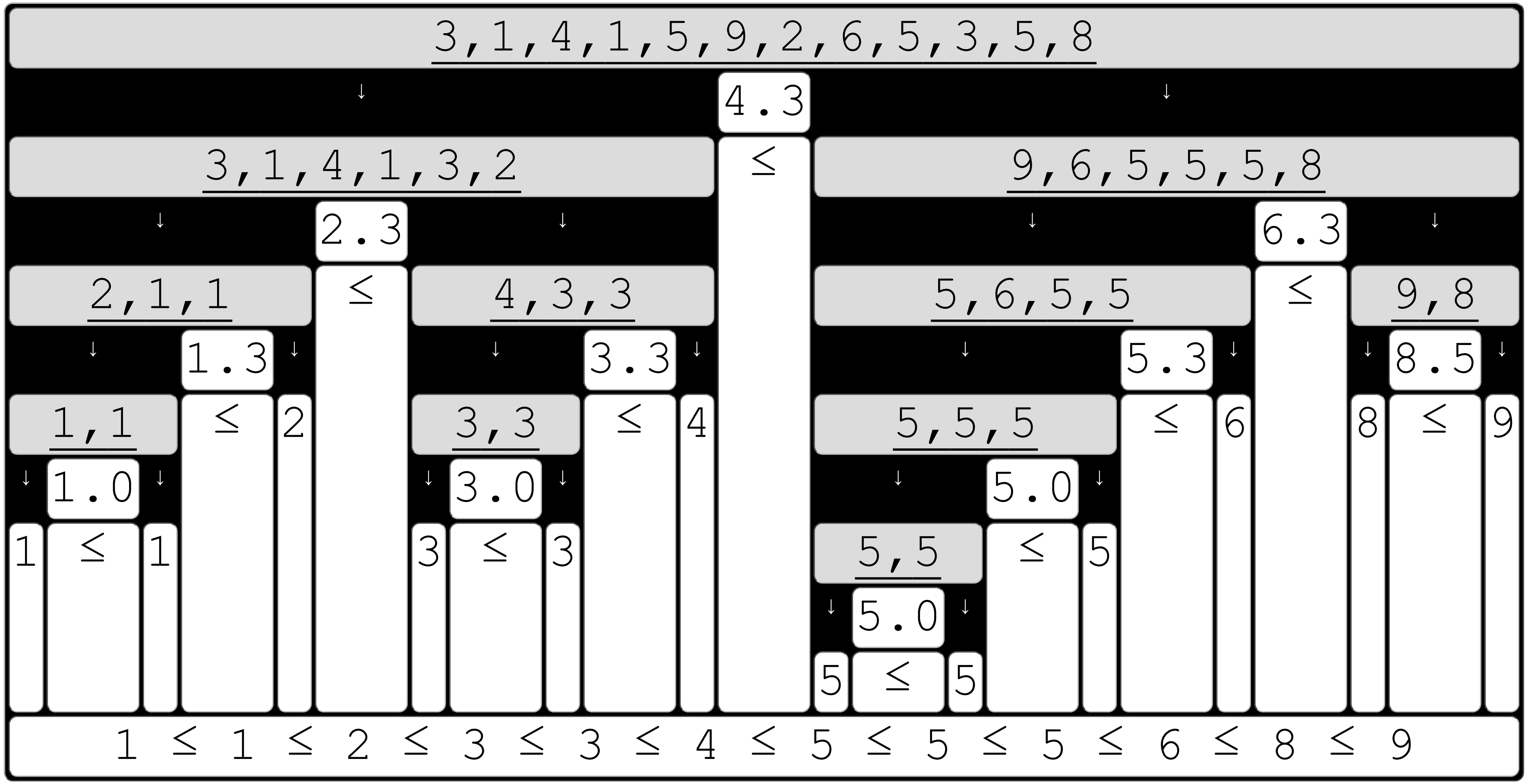
1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

previous

first

Quicksort uses the concept of pivots to split each list near the middle.

next



3,1,4,1,5,9,2,6,5,3,5,8

4.3

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

2.3 6.3

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

1.3 3.3 5.3 8.5

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

1.0 3.0 5.0

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

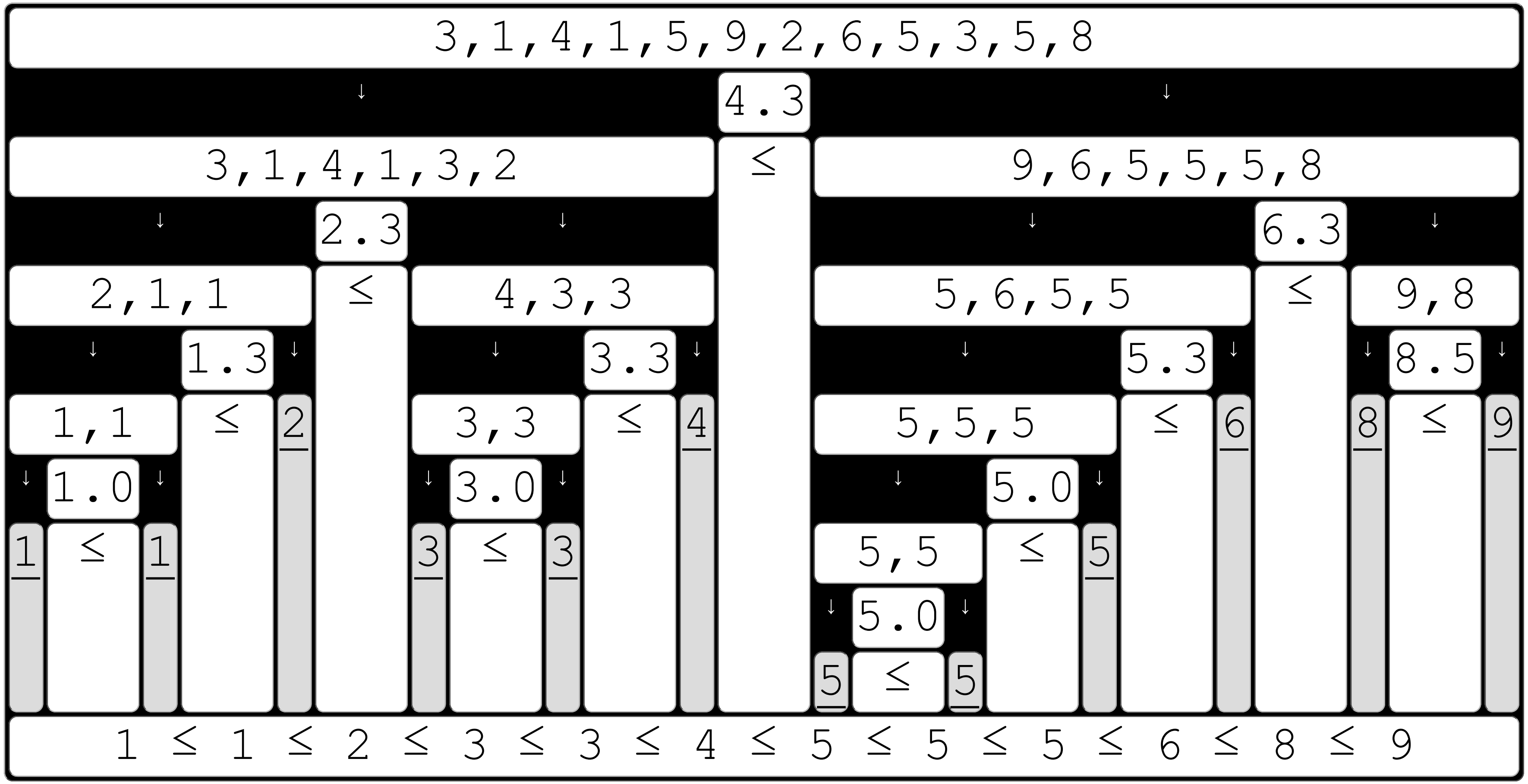
5.0

5 ≤ 5

1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

Quicksort uses the concept of pivots to split each list near the middle.

previous
first
next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

↓ 5.0 ↓

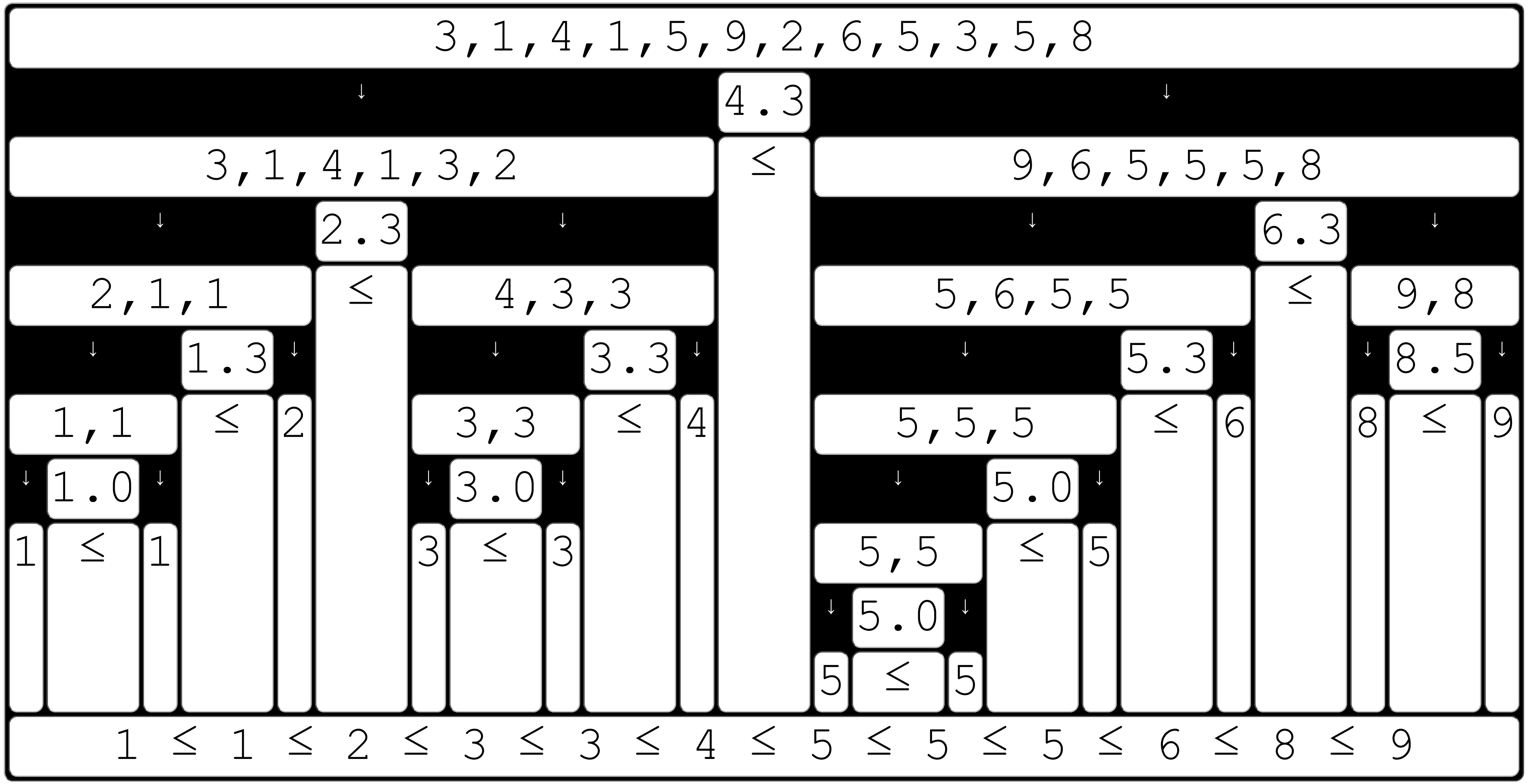
5 ≤ 5

1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

previous
first

This procedure is repeated until each list has only one value (as described in the next steps).

next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

↓ 5.0 ↓

5 ≤ 5

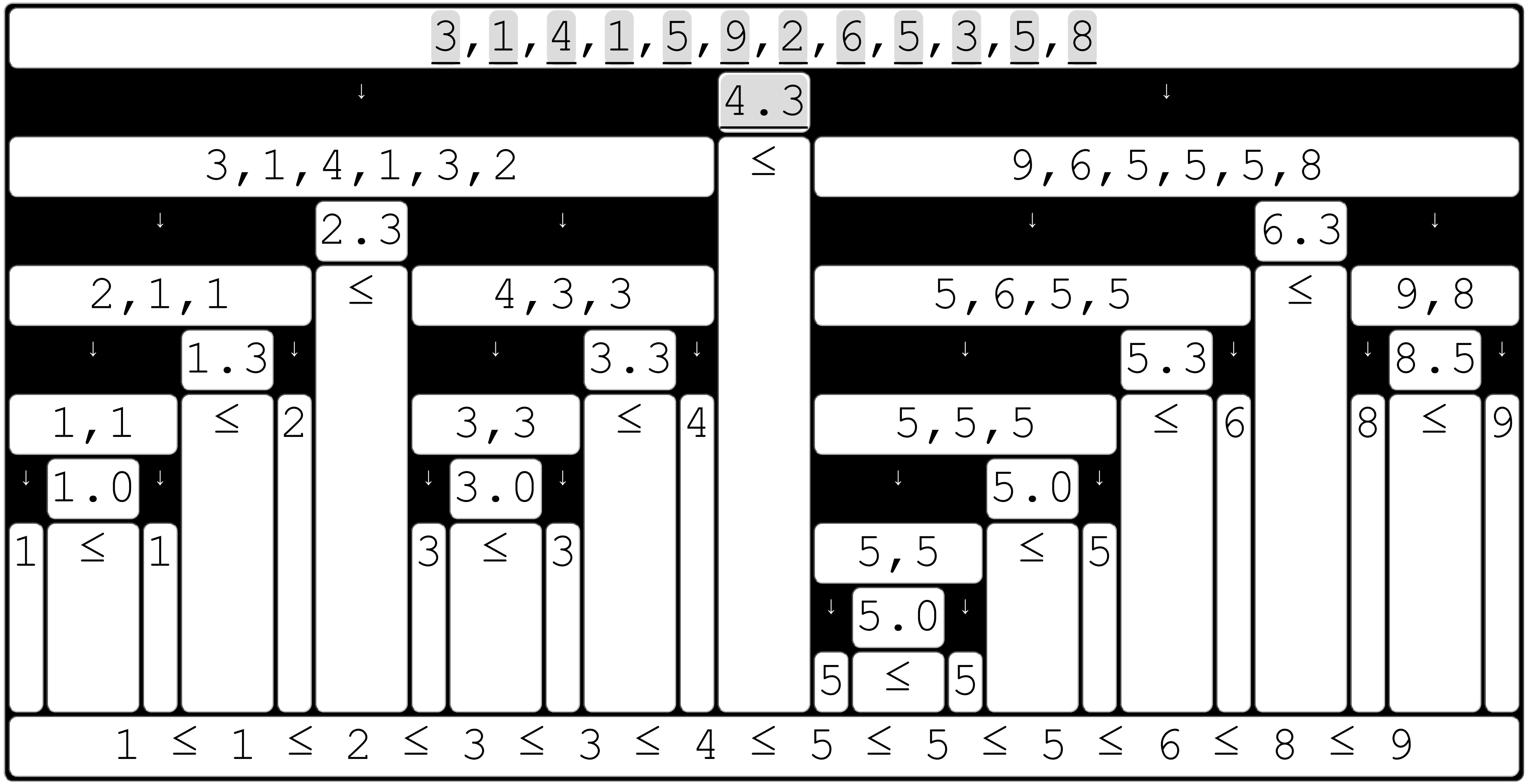
1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

(2) Details of Quicksort.

previous

first

next

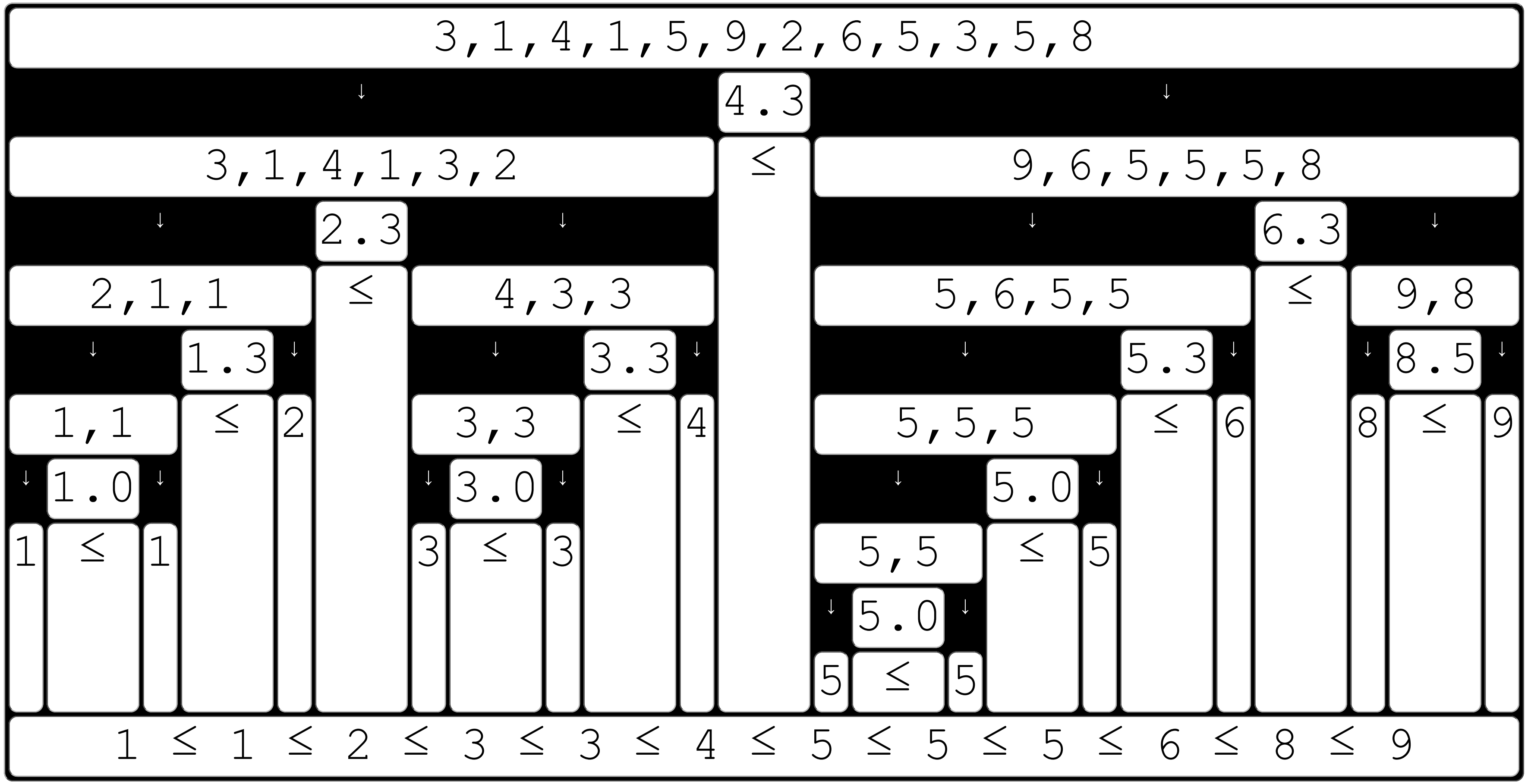


$$4.3 = (3+1+4+1+5+9+2+6+5+3+5+8)/12$$

previous
first

For didactic purposes, the pivot is defined as average (of values of a list). For example: the first pivot is the average of the initial list.

next


3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

↓ 5.0 ↓

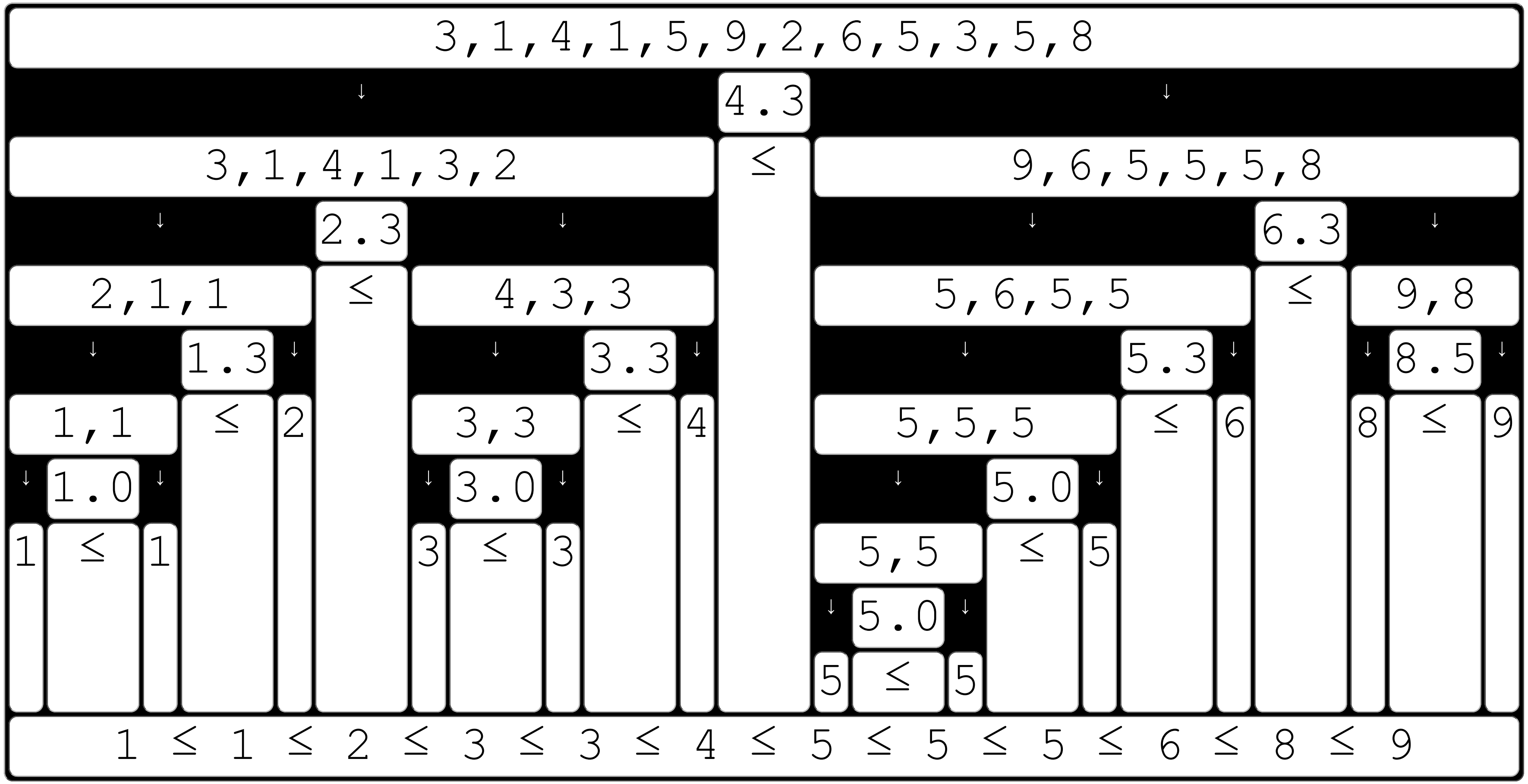
5 ≤ 5

1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

previous
first

Usually, the average is not able to split the list exactly in the middle, but it gets something near that.

next

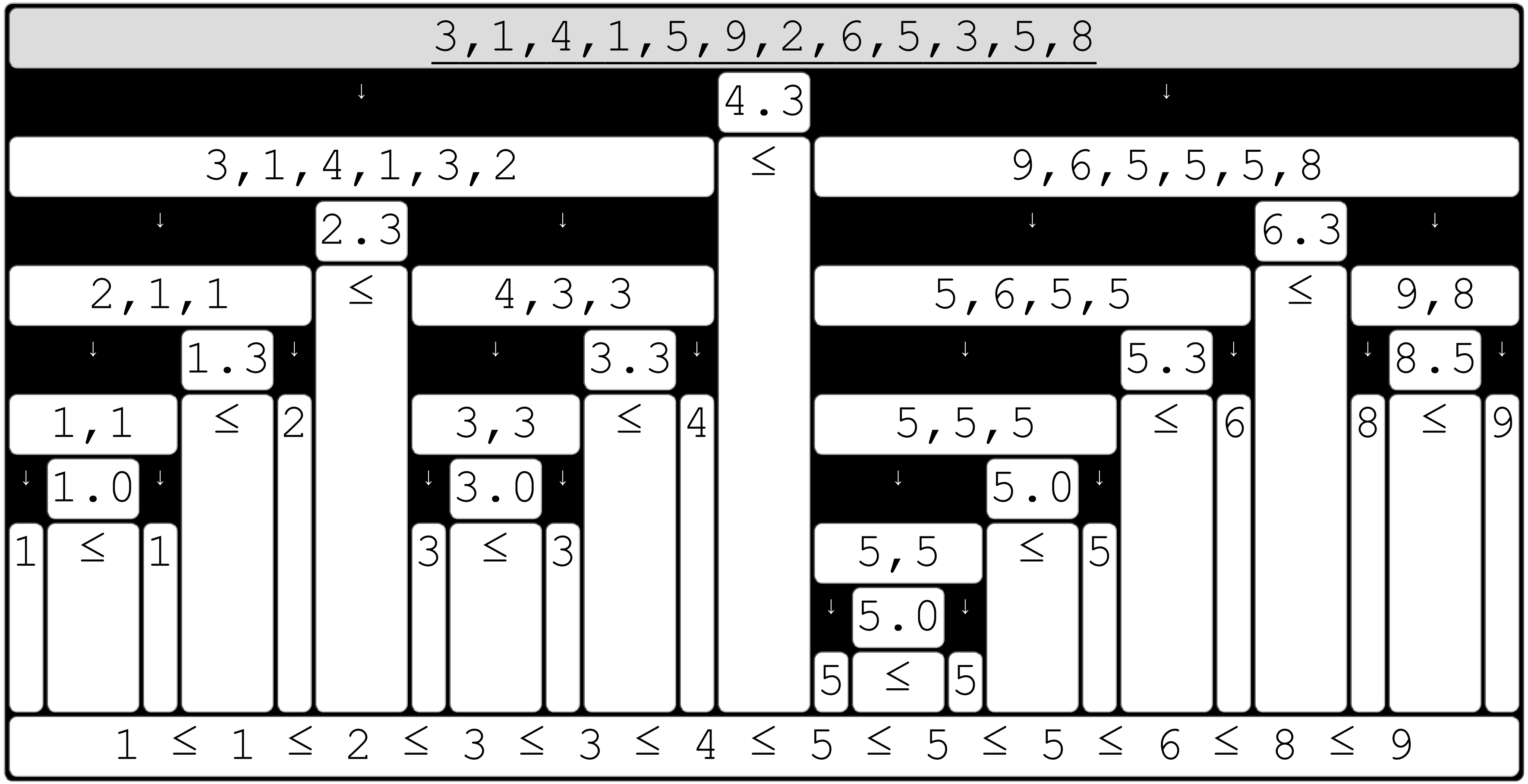


previous

first

At the first level, the initial list is splitted (around the pivot) which results in two lists.

next

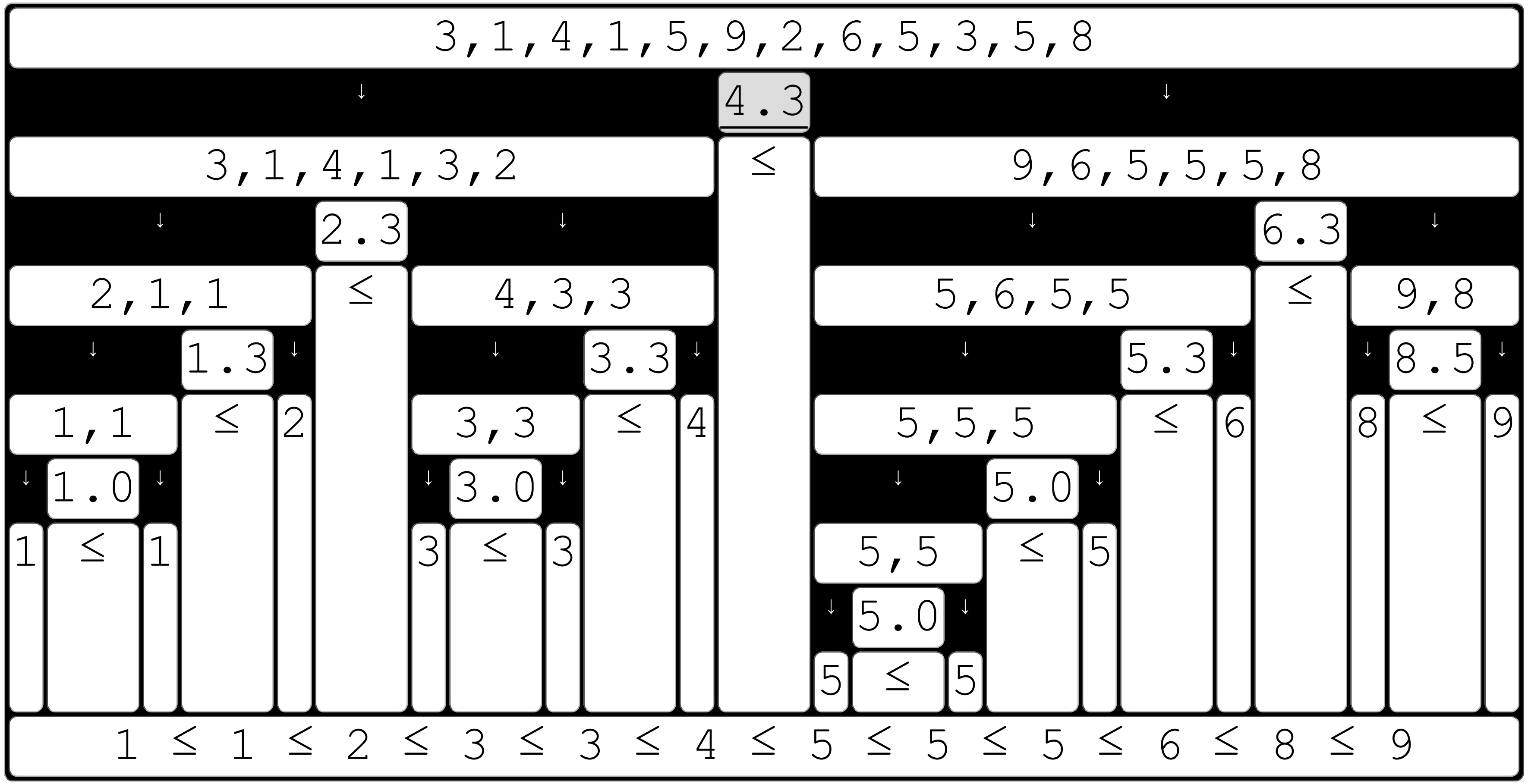


previous

first

At the first level, the initial list is splitted (around the pivot) which results in two lists.

next

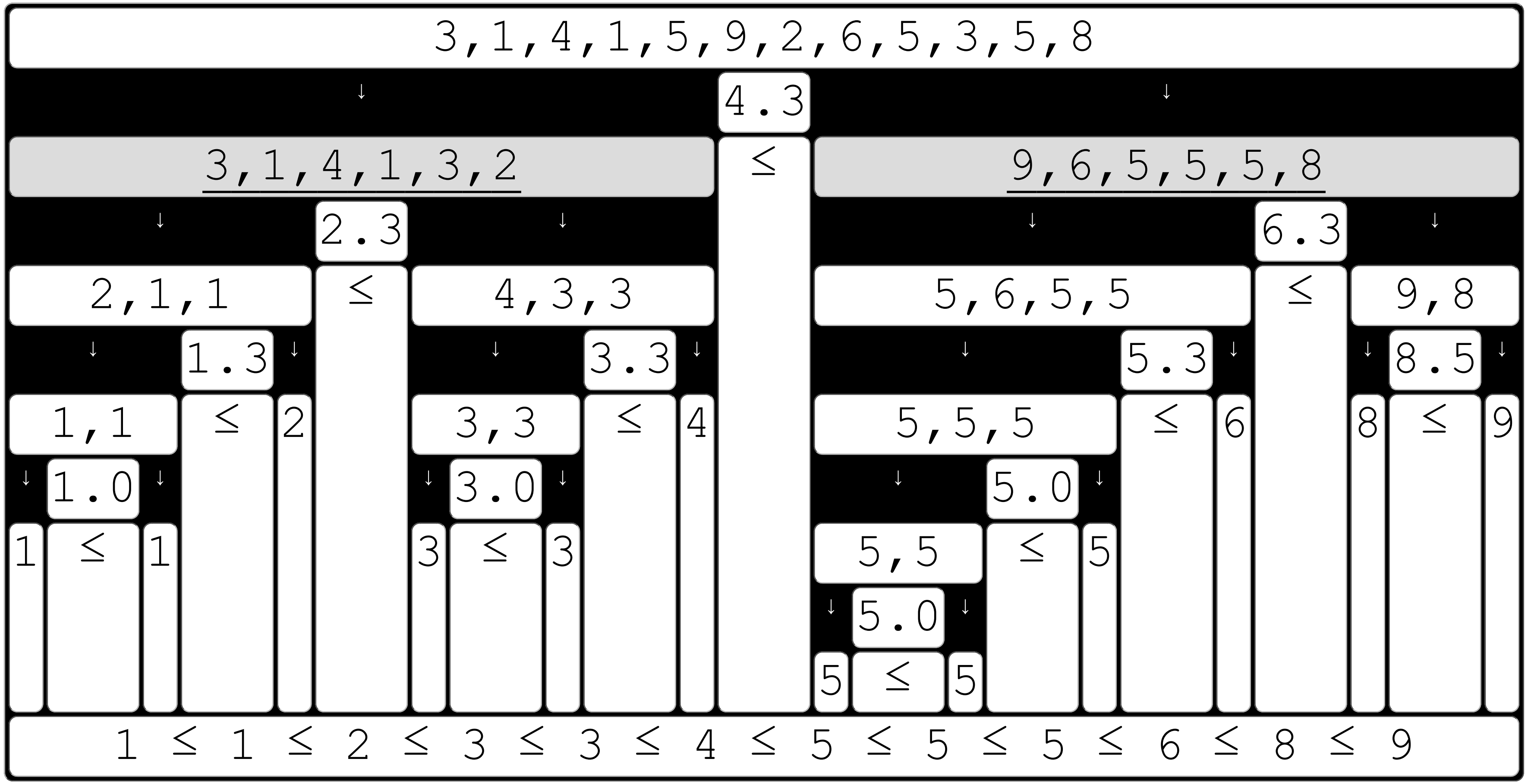


previous

first

At the first level, the initial list is splitted (around the pivot) which results in two lists.

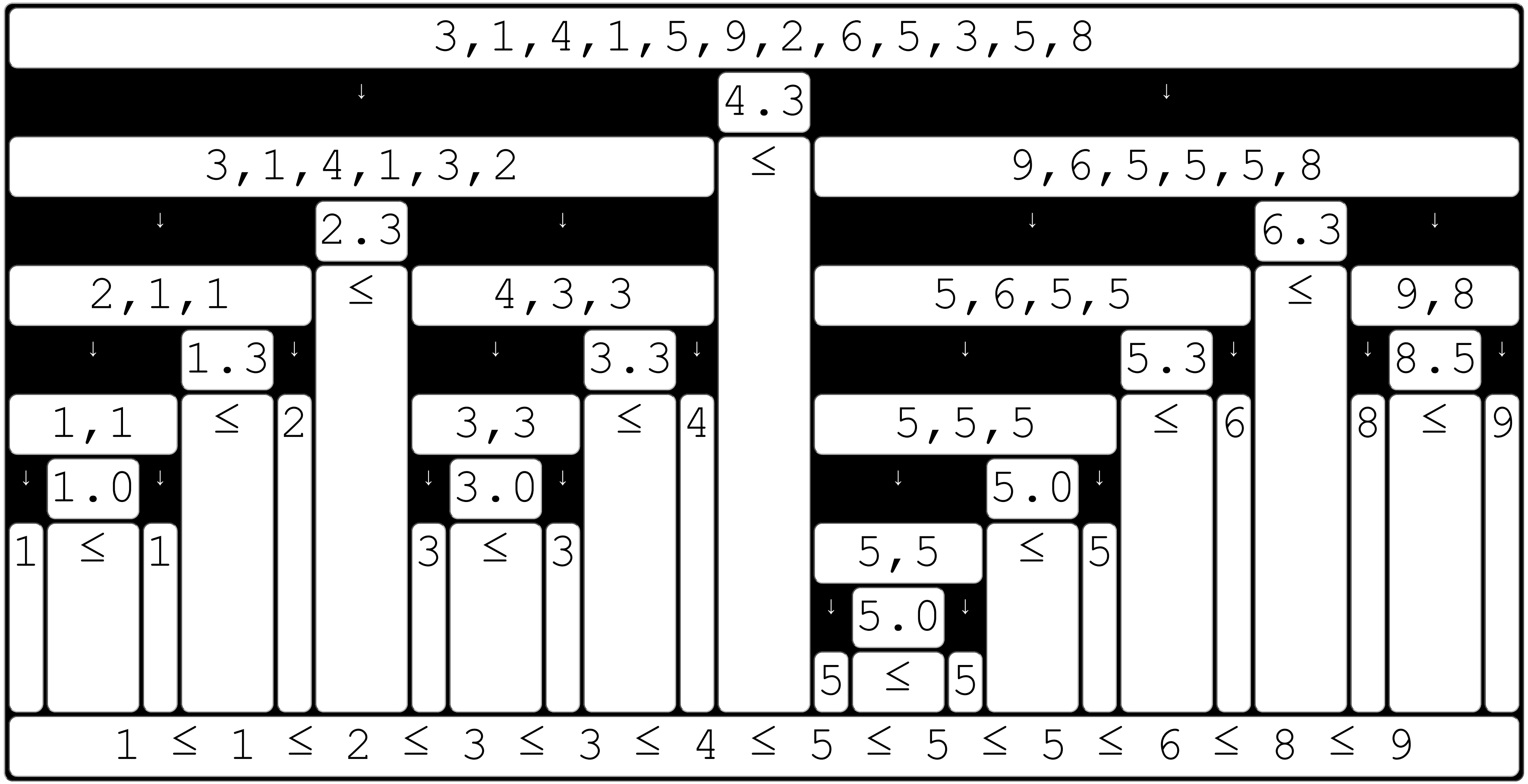
next



previous
first

At the first level, the initial list is splitted (around the pivot) which results in two lists.

next

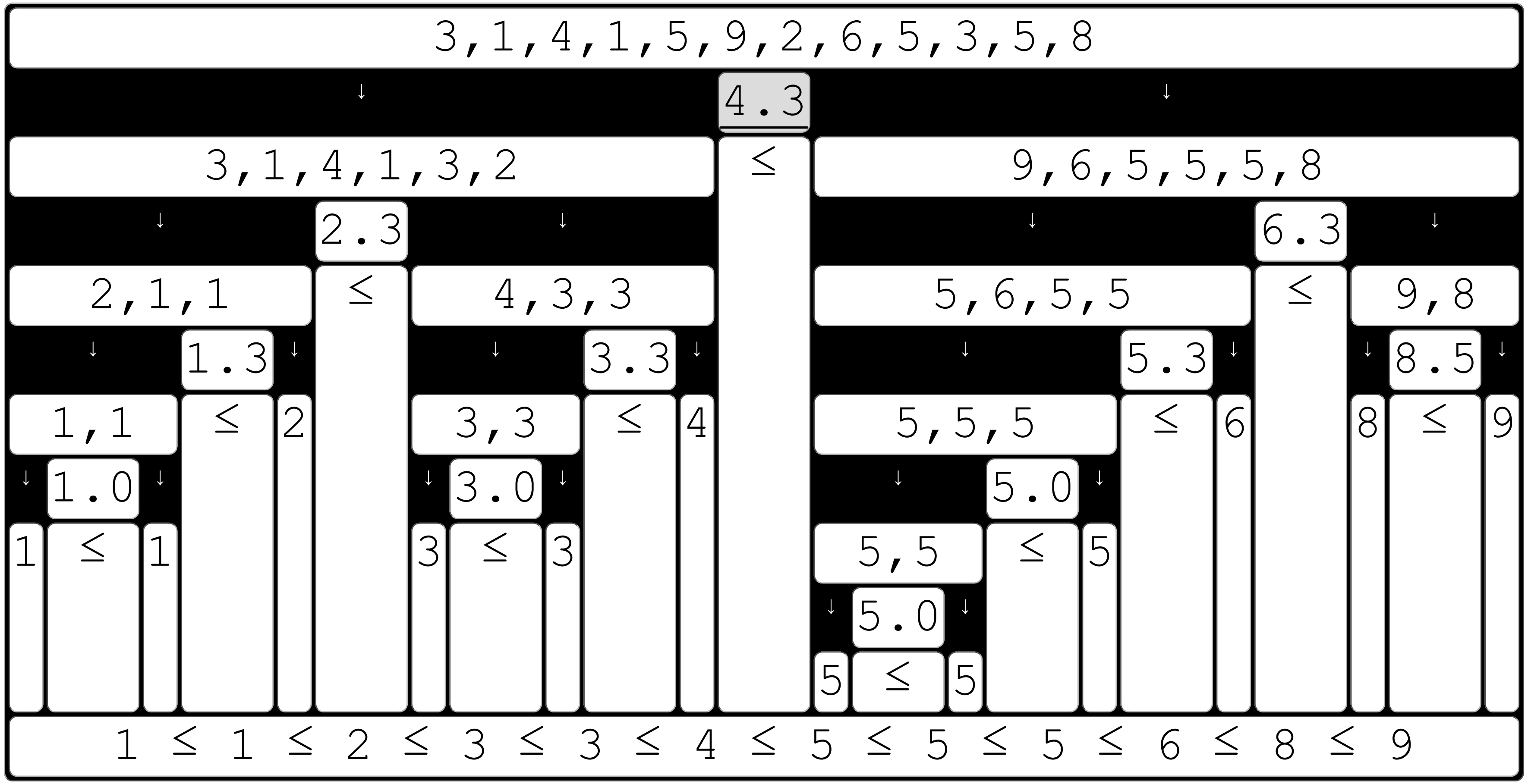


previous

first

Compared to pivot those two lists have: values smaller than the pivot, and values larger than the pivot.

next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

↓ 5.0 ↓

5 ≤ 5

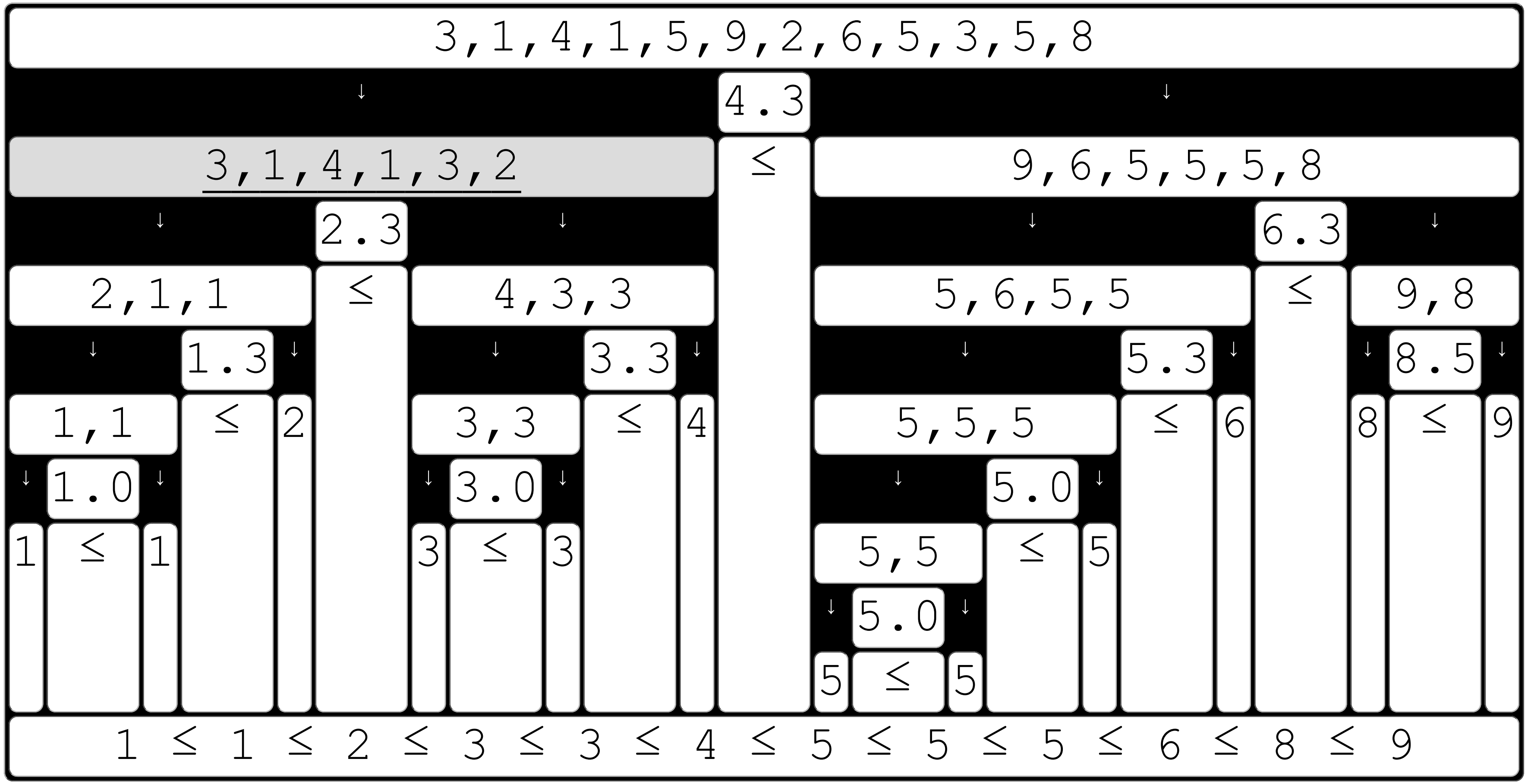
1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

previous

first

Compared to pivot those two lists have: values smaller than the pivot, and values larger than the pivot.

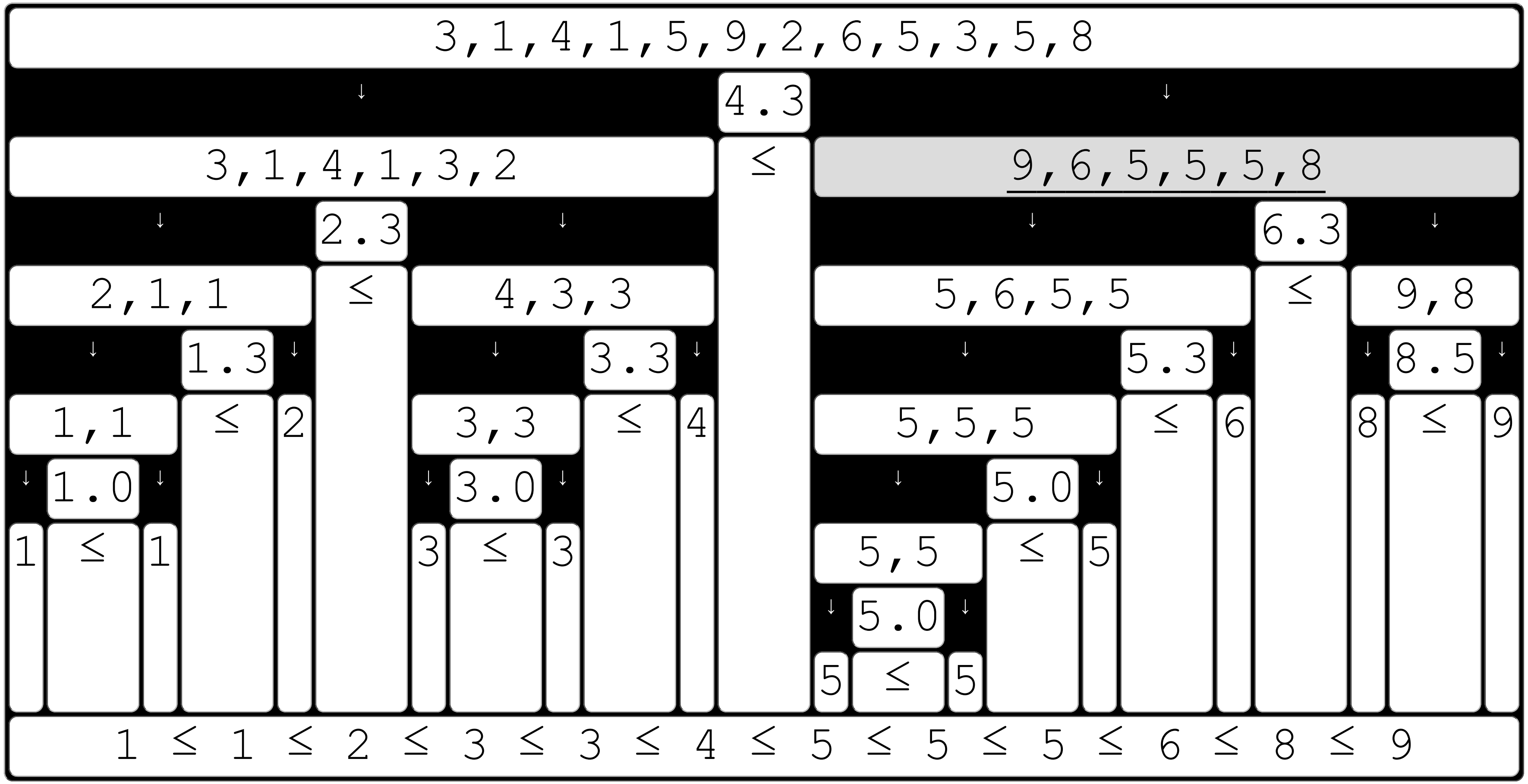
next



previous
first

Compared to pivot those two lists have: values smaller than the pivot, and values larger than the pivot.

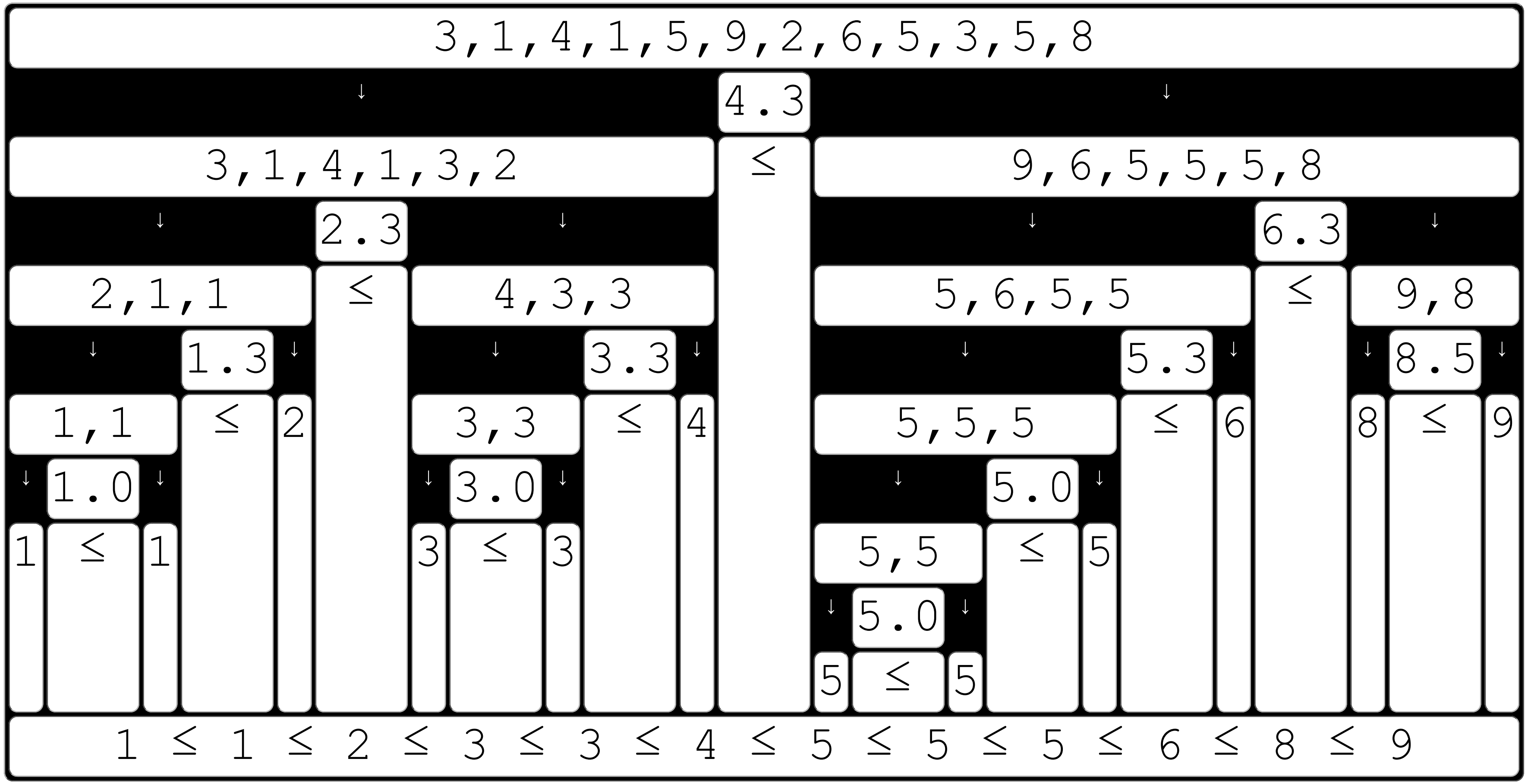
next



previous
first

Compared to pivot those two lists have: values smaller than the pivot, and values larger than the pivot.

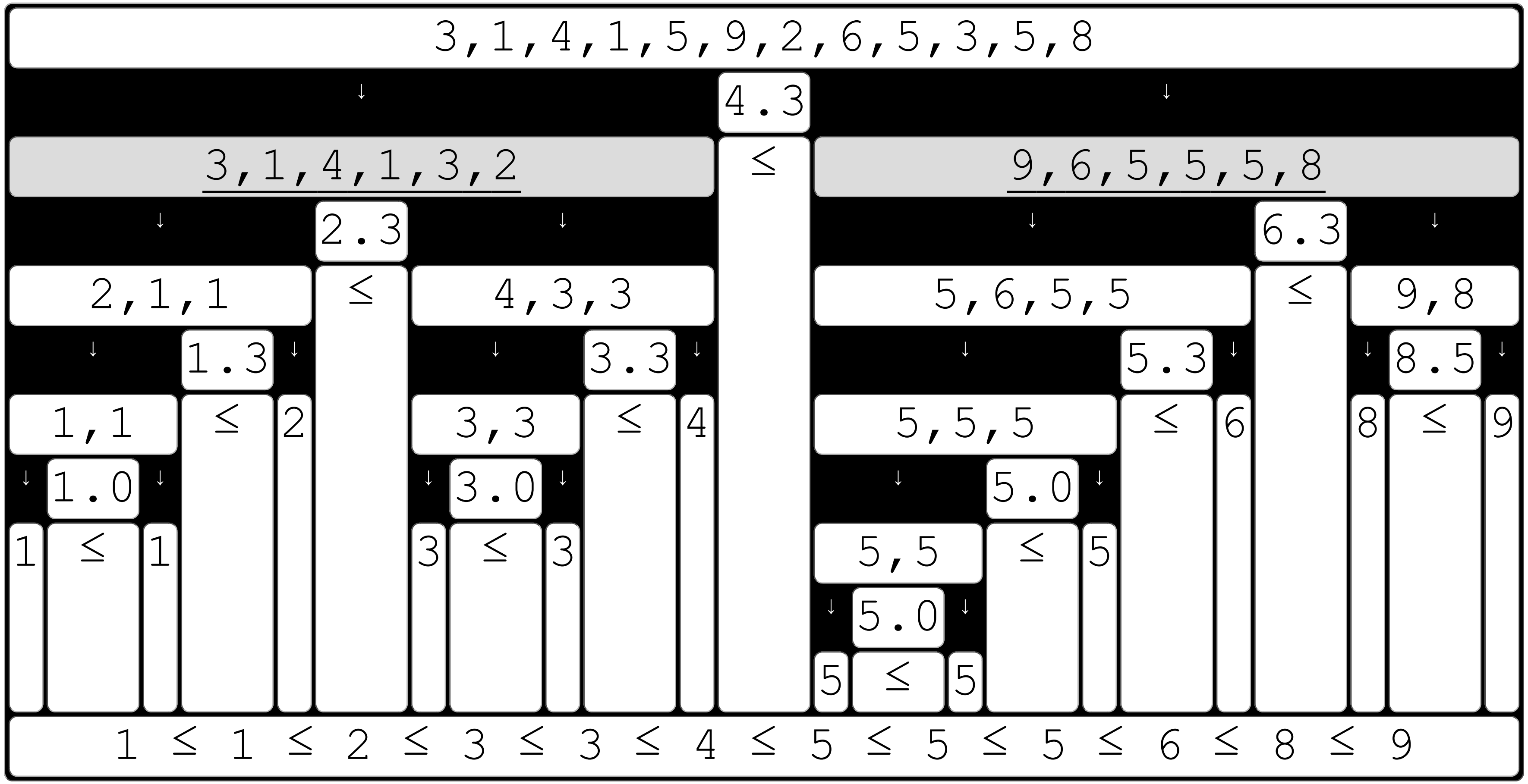
next



previous
first

At the second level, those two lists require their own pivots (that means: two pivots). The split procedure results in four lists.

next

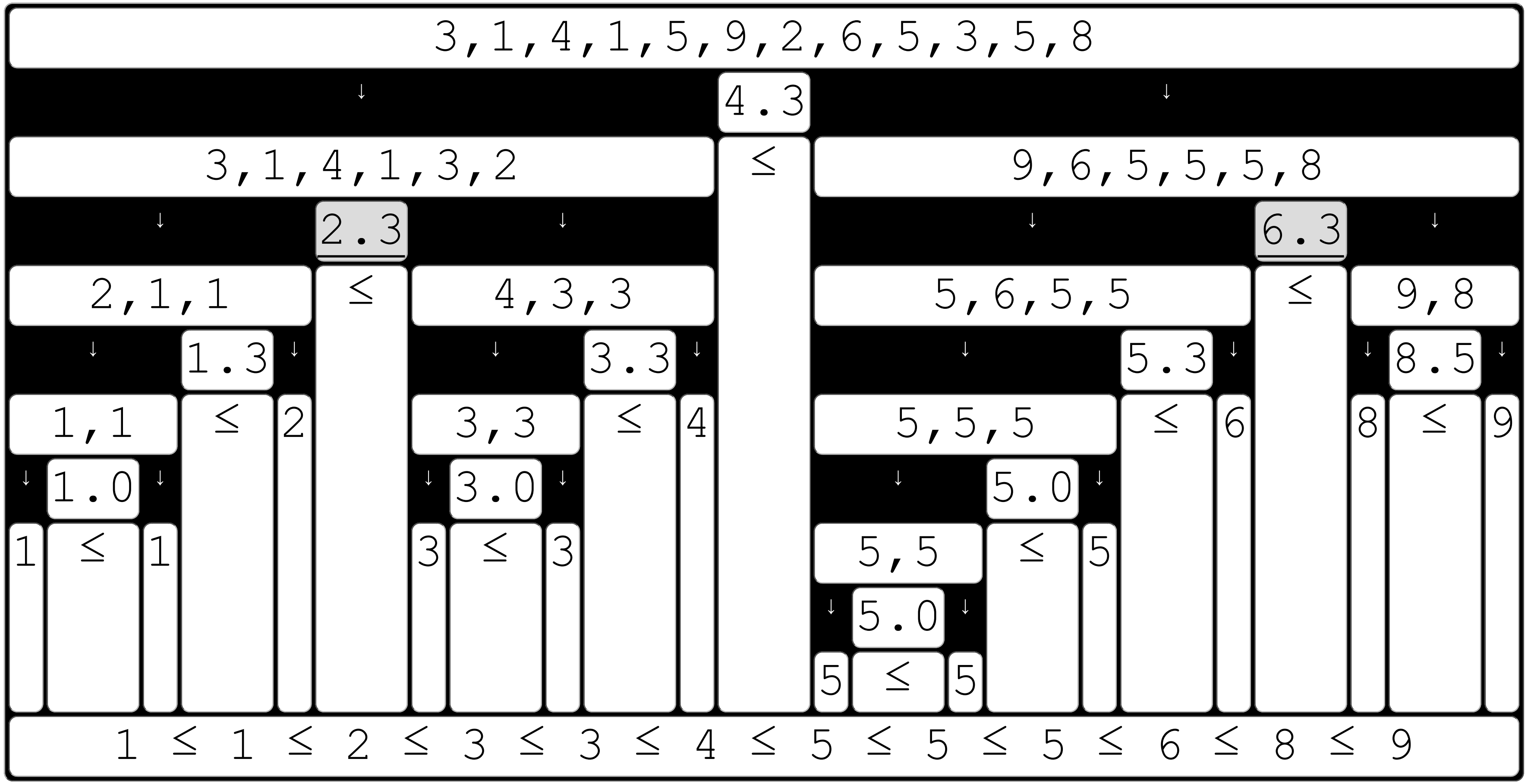


previous

first

At the second level, those two lists require their own pivots (that means: two pivots). The split procedure results in four lists.

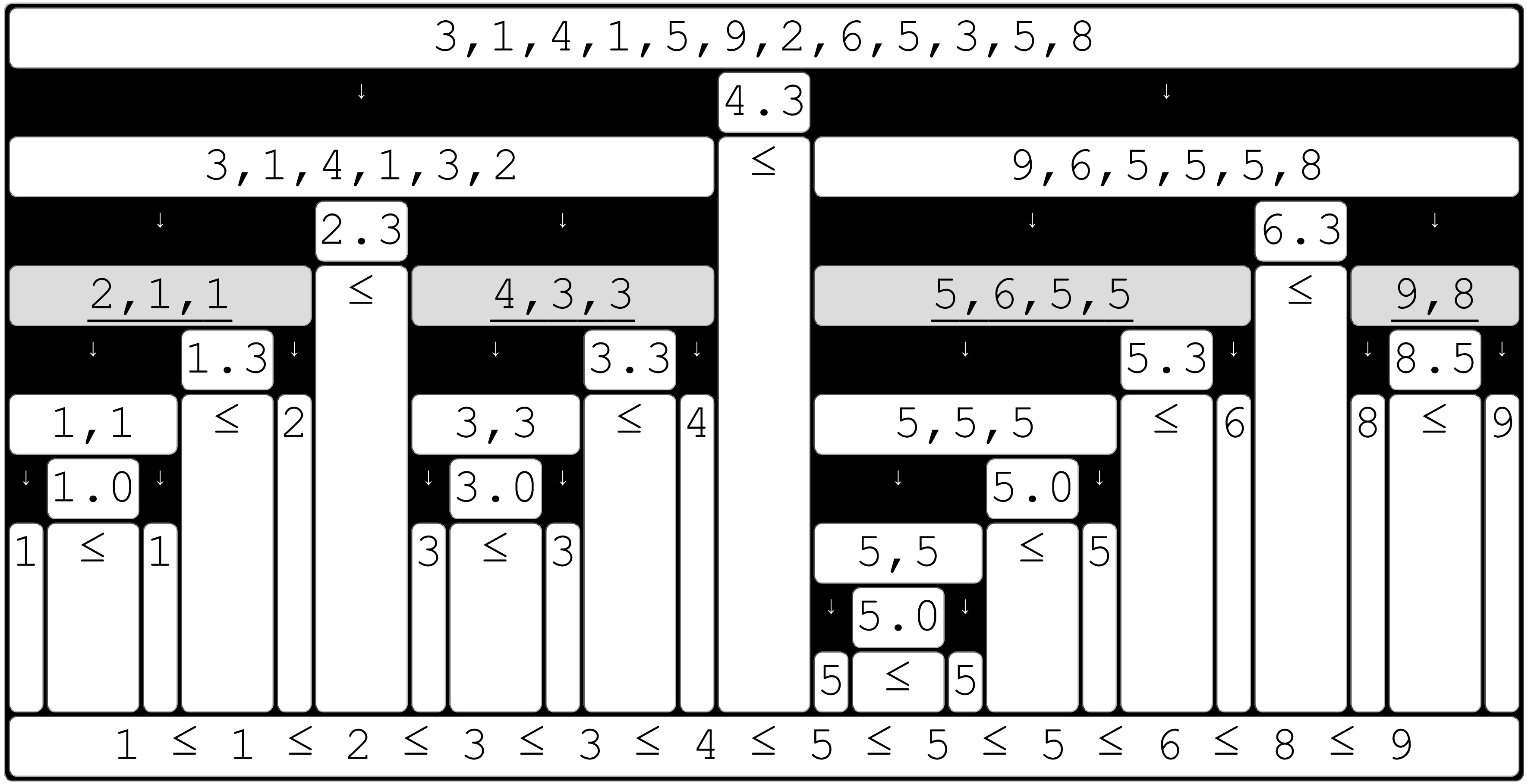
next



previous
first

At the second level, those two lists require their own pivots (that means: two pivots). The split procedure results in four lists.

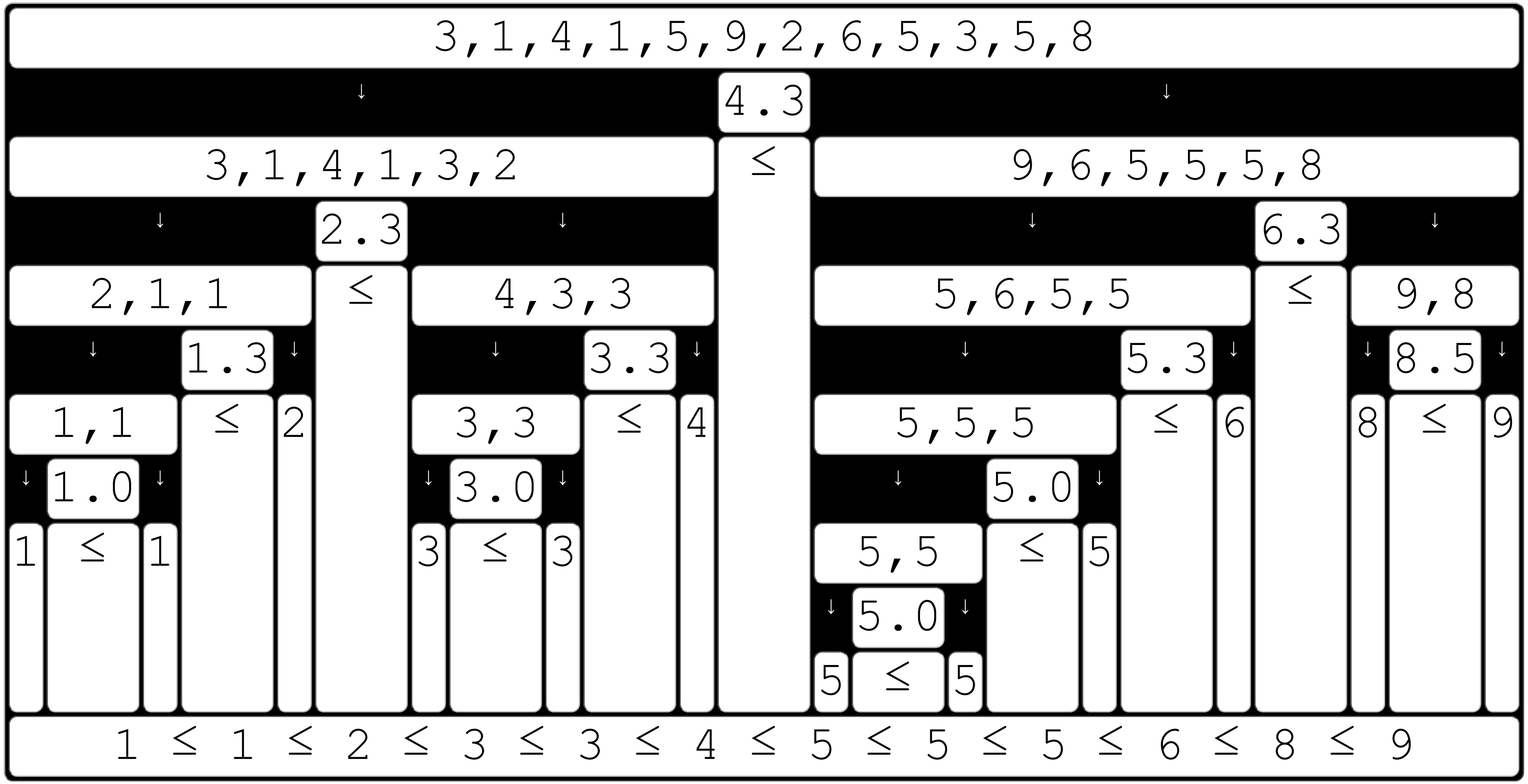
next



previous
first

At the second level, those two lists require their own pivots (that means: two pivots). The split procedure results in four lists.

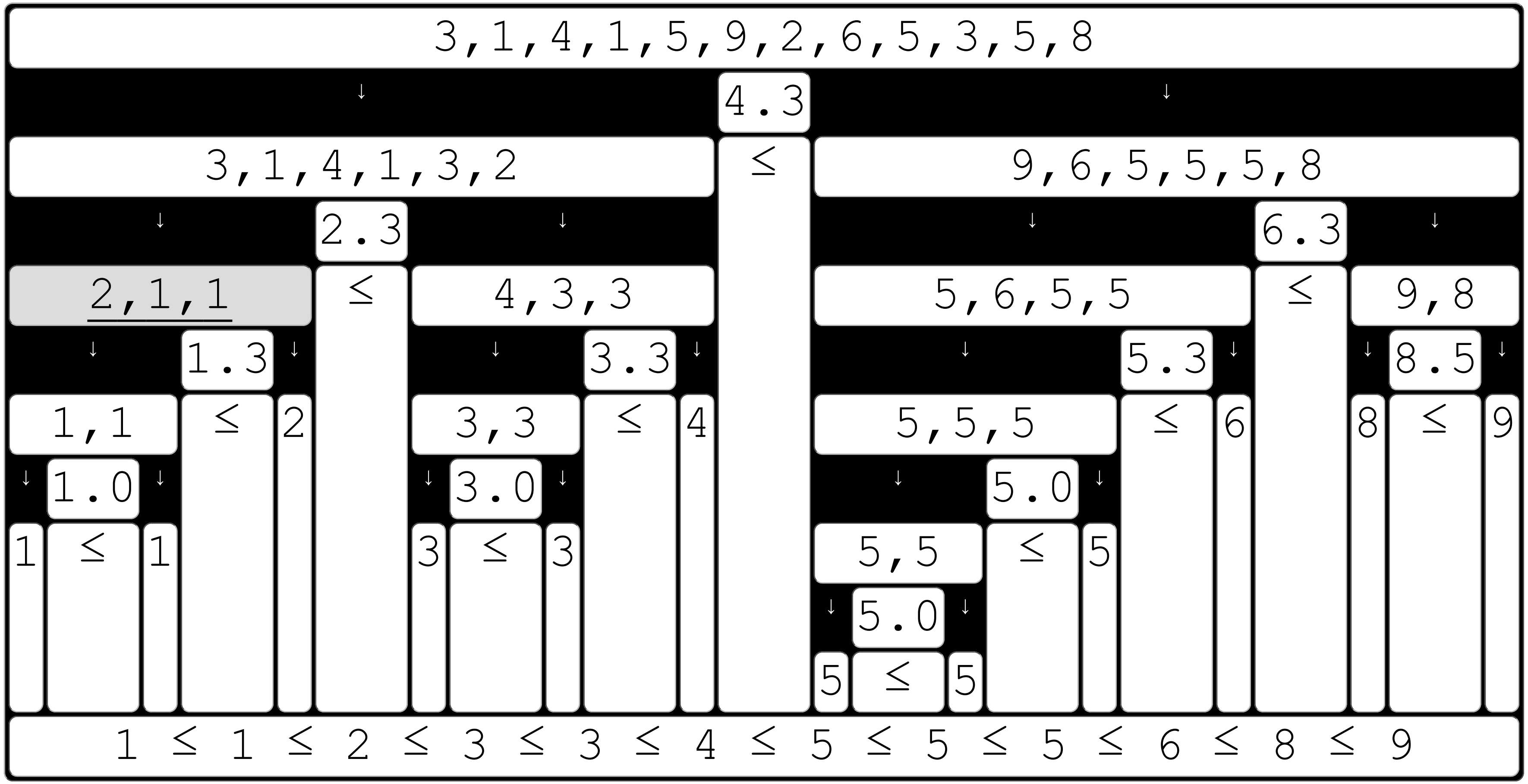
next



previous
first

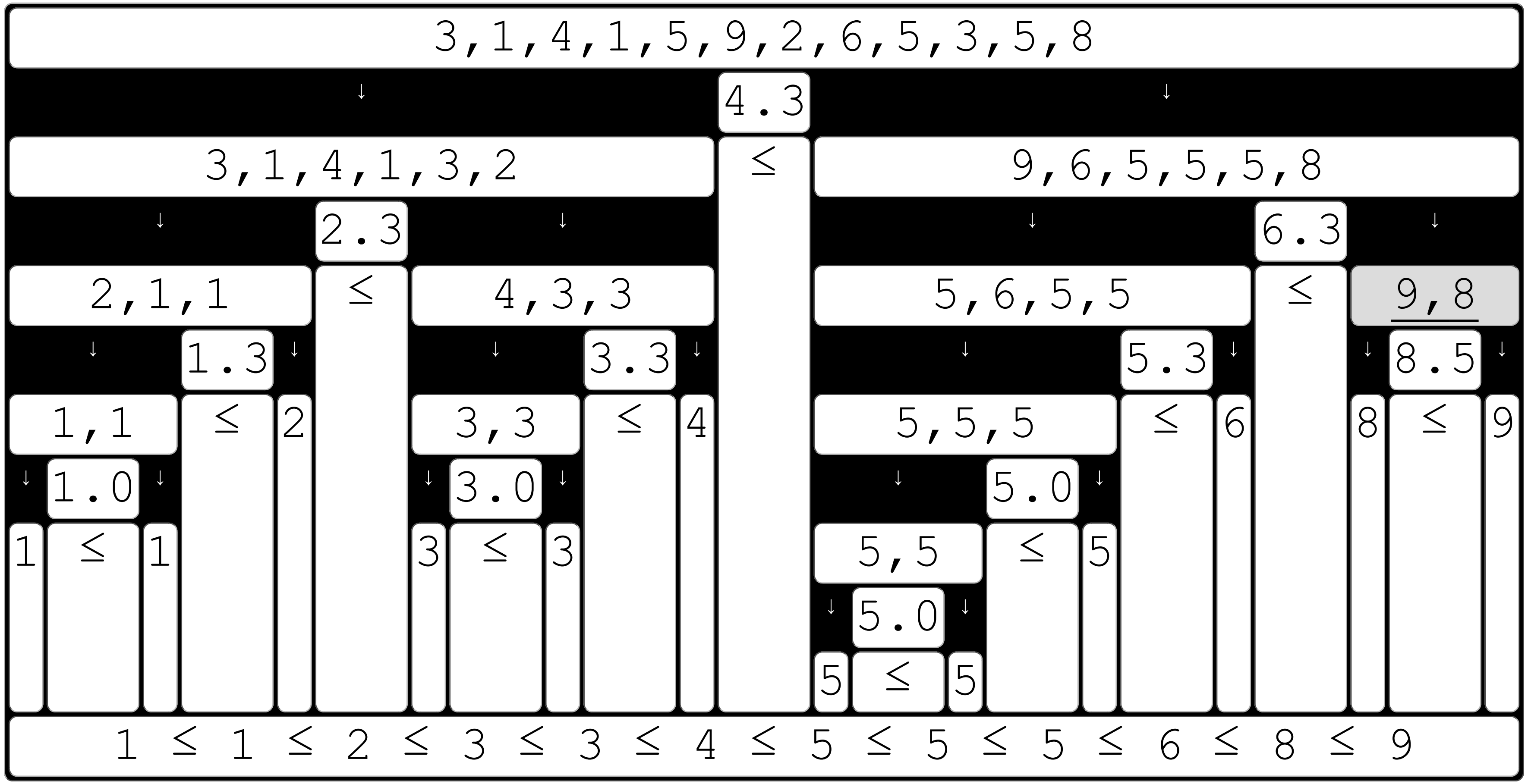
The list at the left side has the smallest values, the list at the right side has the largest values, and everything else in between.

next



The list at the left side has the smallest values, the list at the right side has the largest values, and everything else in between.

previous
first
next

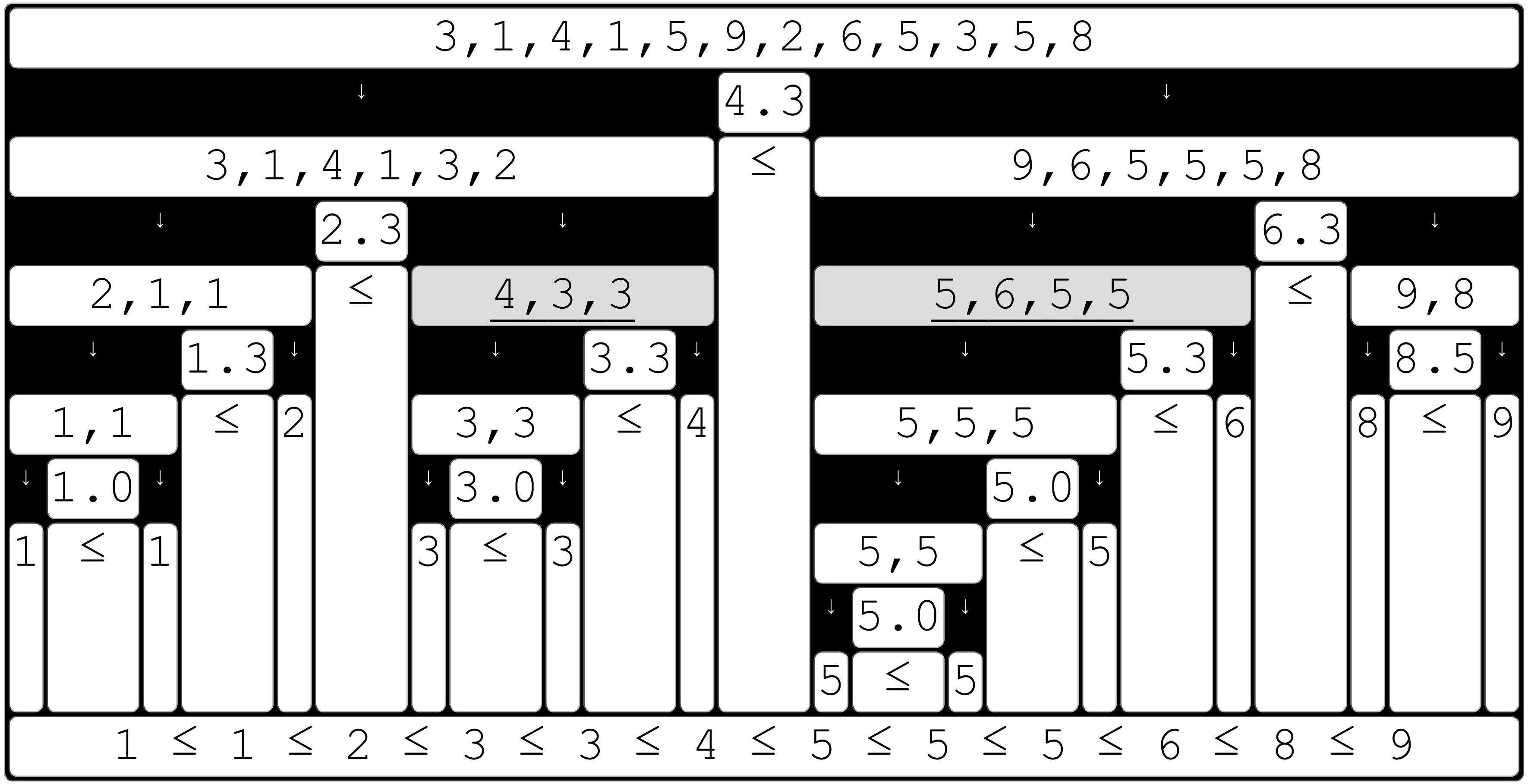


previous

first

The list at the left side has the smallest values, the list at the right side has the largest values, and everything else in between.

next

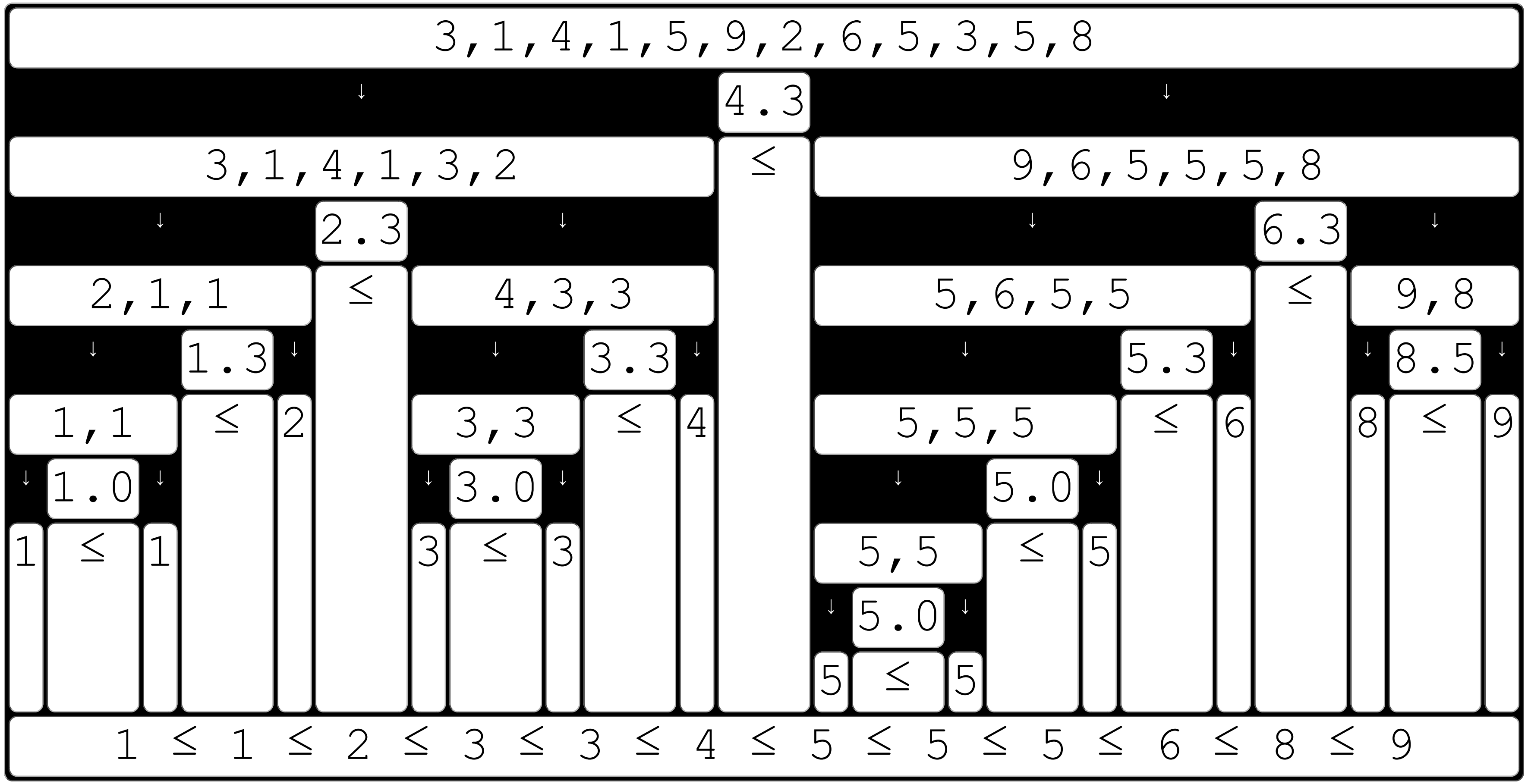


previous

first

The list at the left side has the smallest values, the list at the right side has the largest values, and everything else in between.

next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

↓ 5.0 ↓

5 ≤ 5

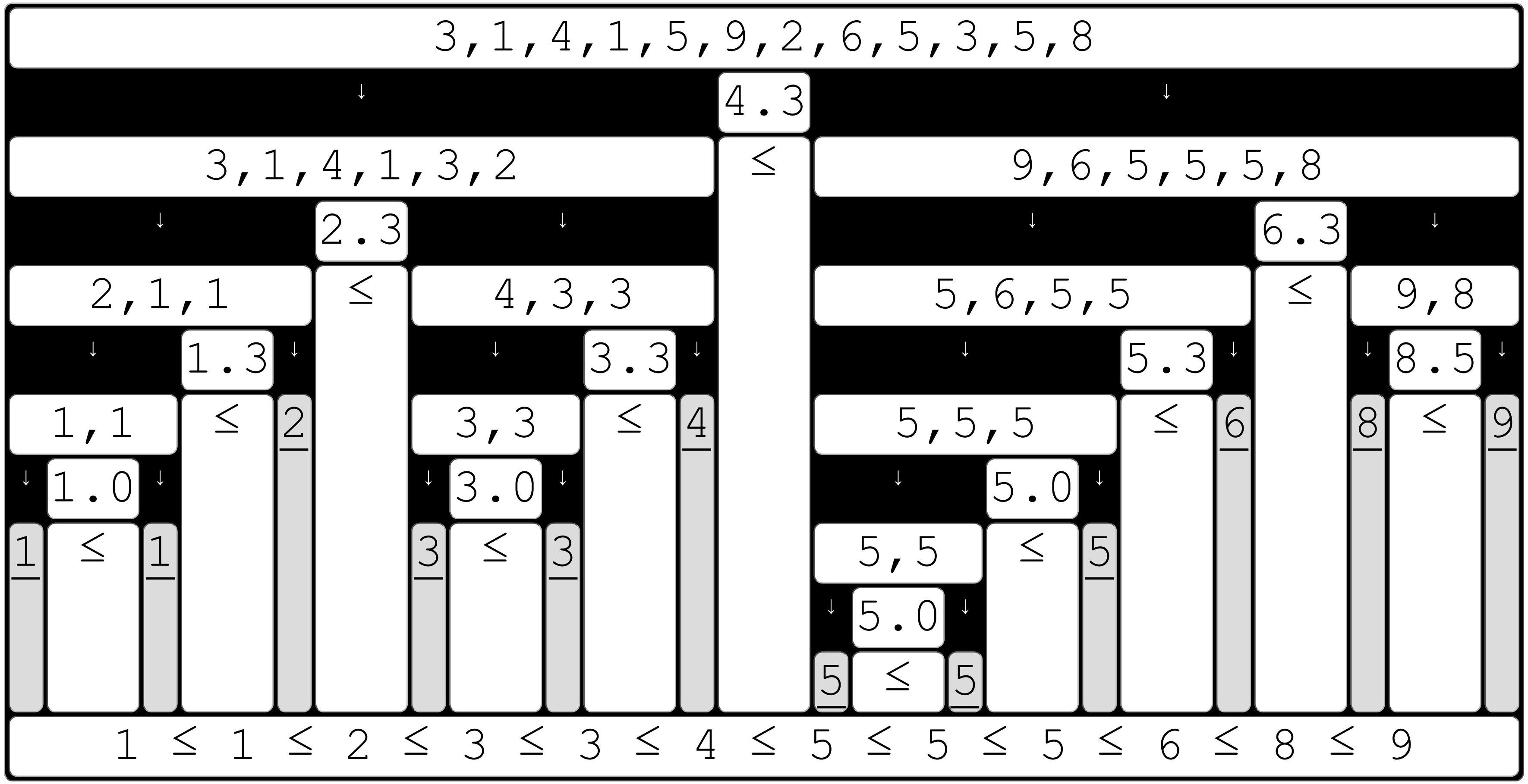
1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

previous

first

The algorithm repeats this procedure until each list has the size of only one value. Those values are the result of the sort algorithm.

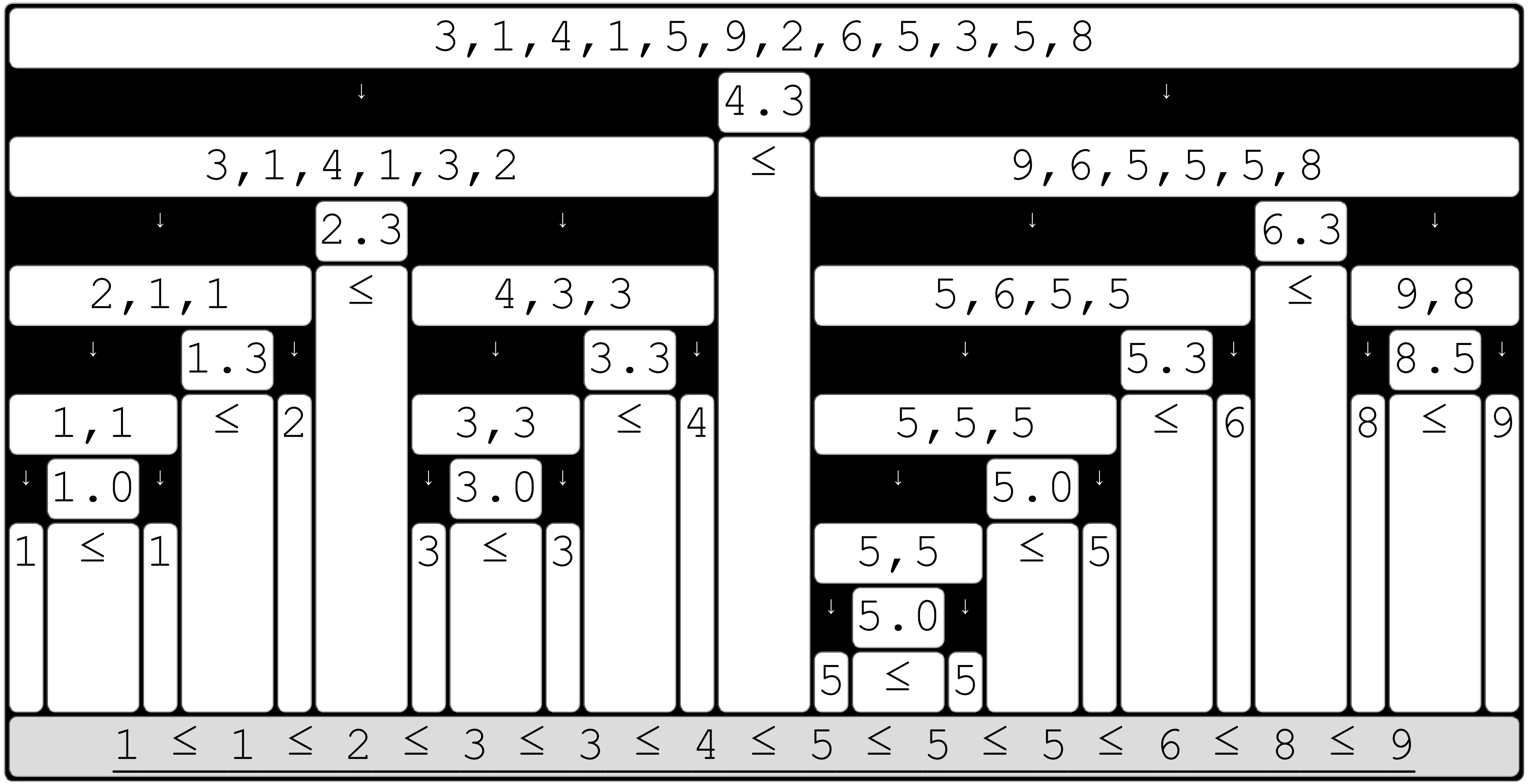
next



previous
first

The algorithm repeats this procedure until each list has the size of only one value. Those values are the result of the sort algorithm.

next

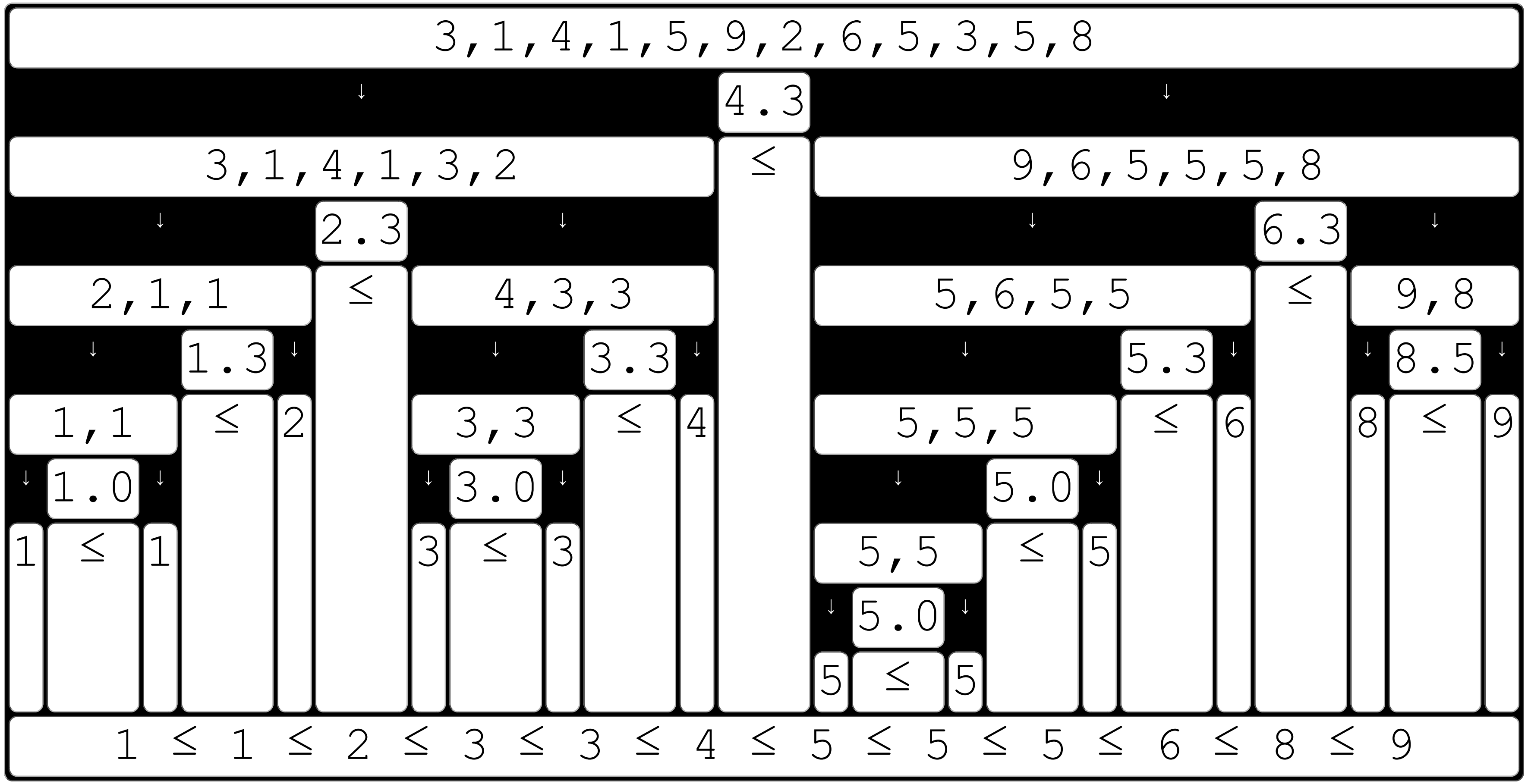


previous

first

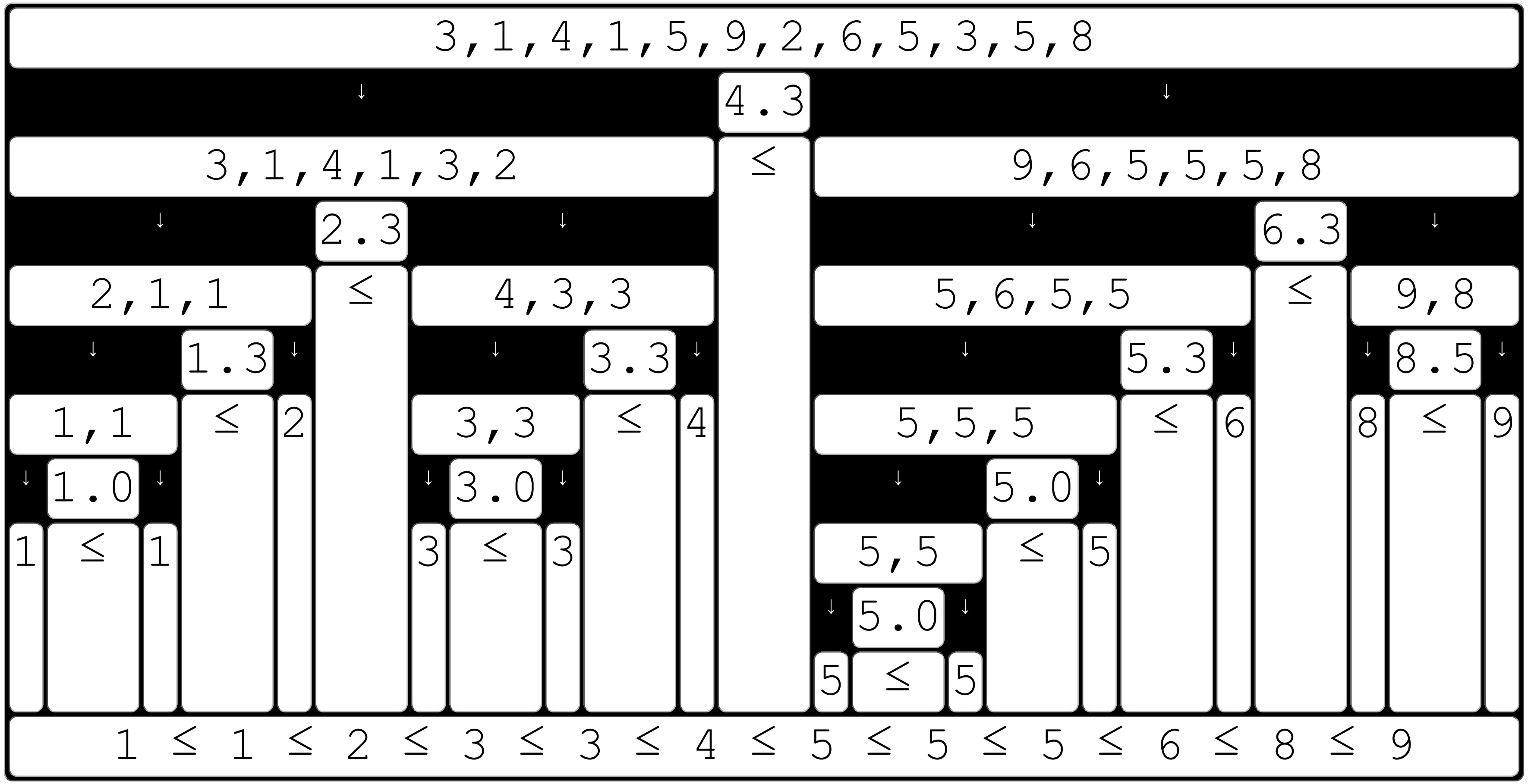
The algorithm repeats this procedure until each list has the size of only one value. Those values are the result of the sort algorithm.

next



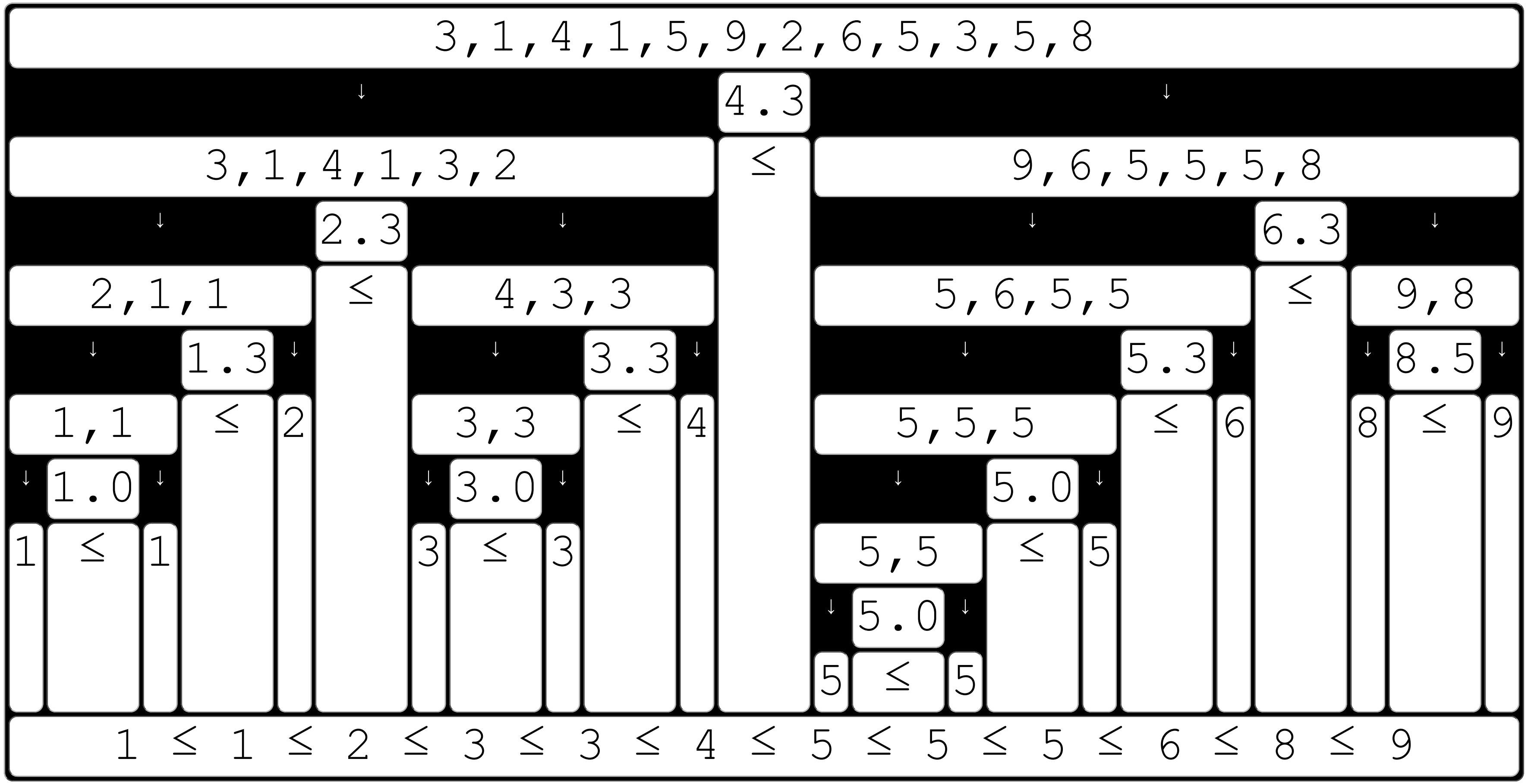
"Recursive Algorithm" is the name of this kind of repetitive procedure.

previous
first
next



(3) Advanced concepts: in-place.

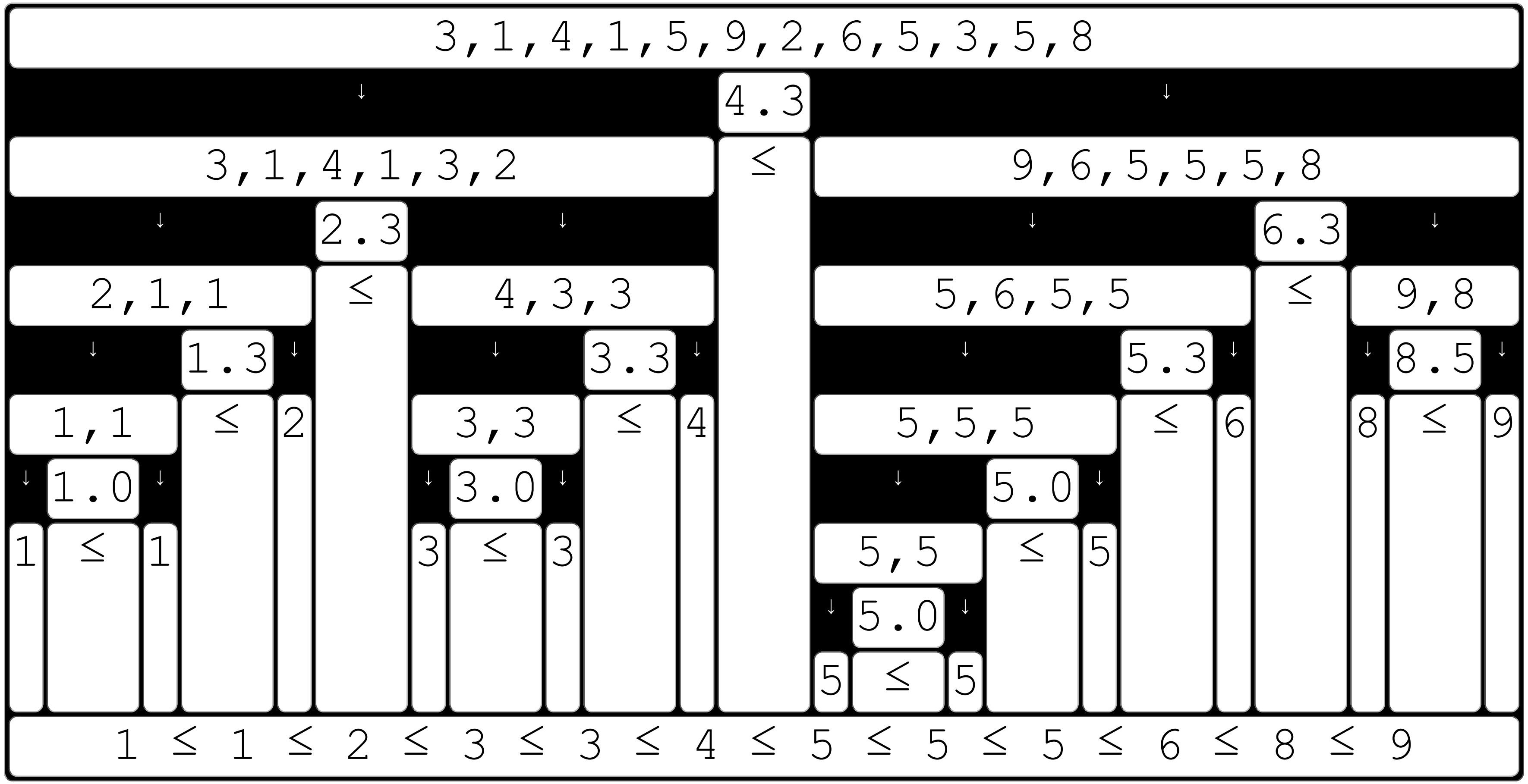
previous
first
next



previous
first

Quicksort is an efficient sort algorithm, and one of the few in-place algorithms (in this category).

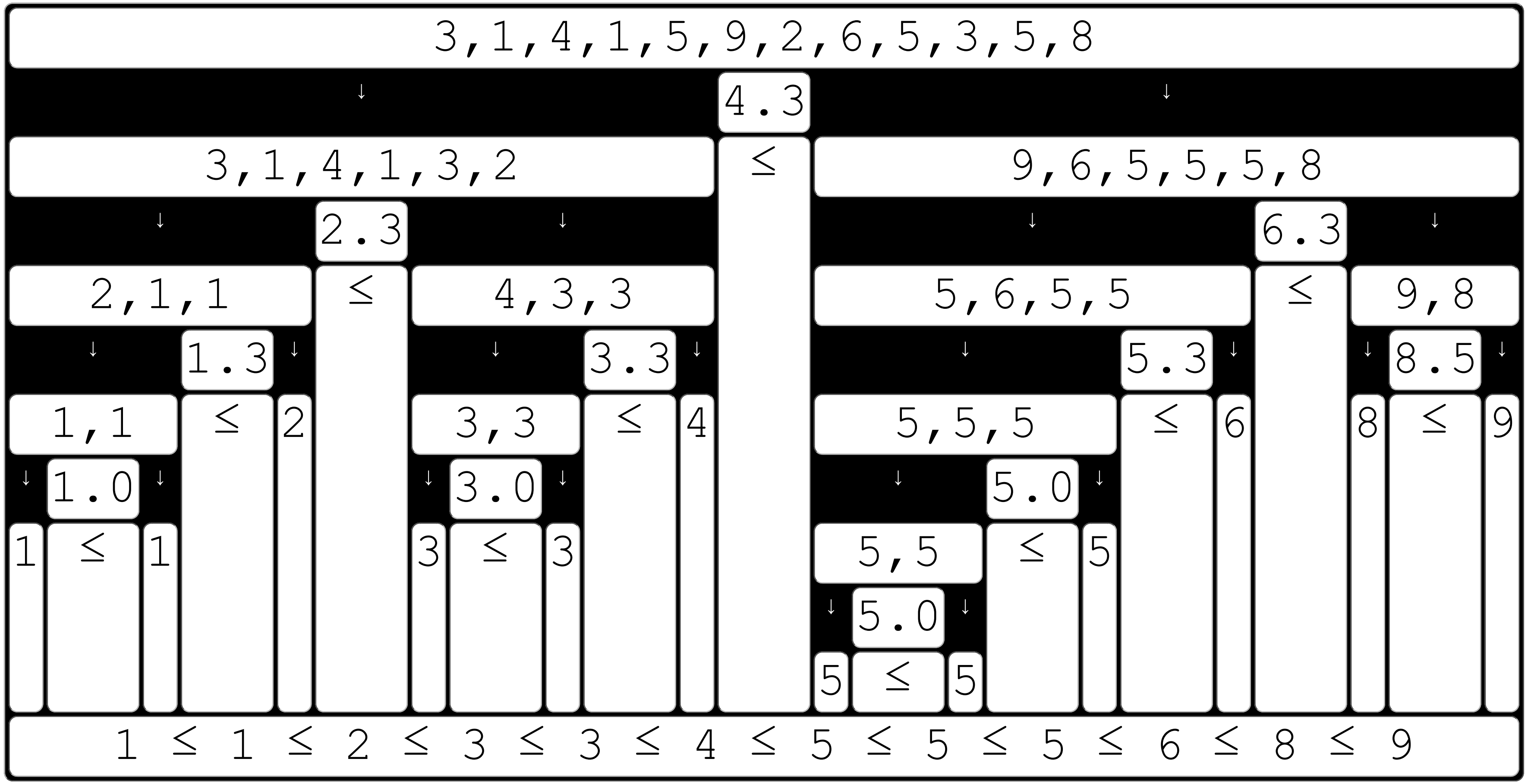
next



previous
first

In-place means: the algorithm does not require additional storage memory. Given two memory positions, Quicksort exchanges its values.

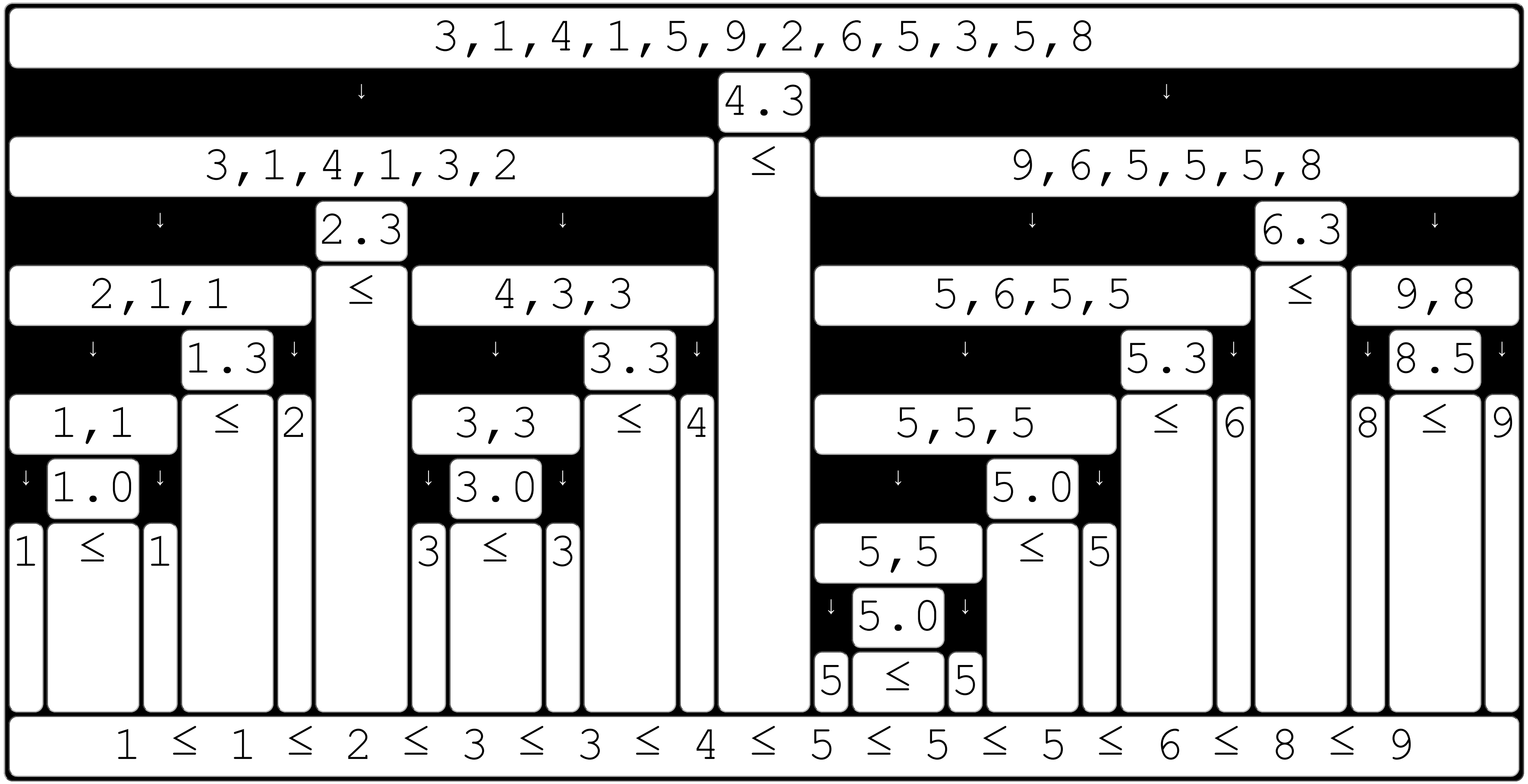
next



previous
first

A naive approach would create (at the split procedure) two lists with the same order of previous iteration.

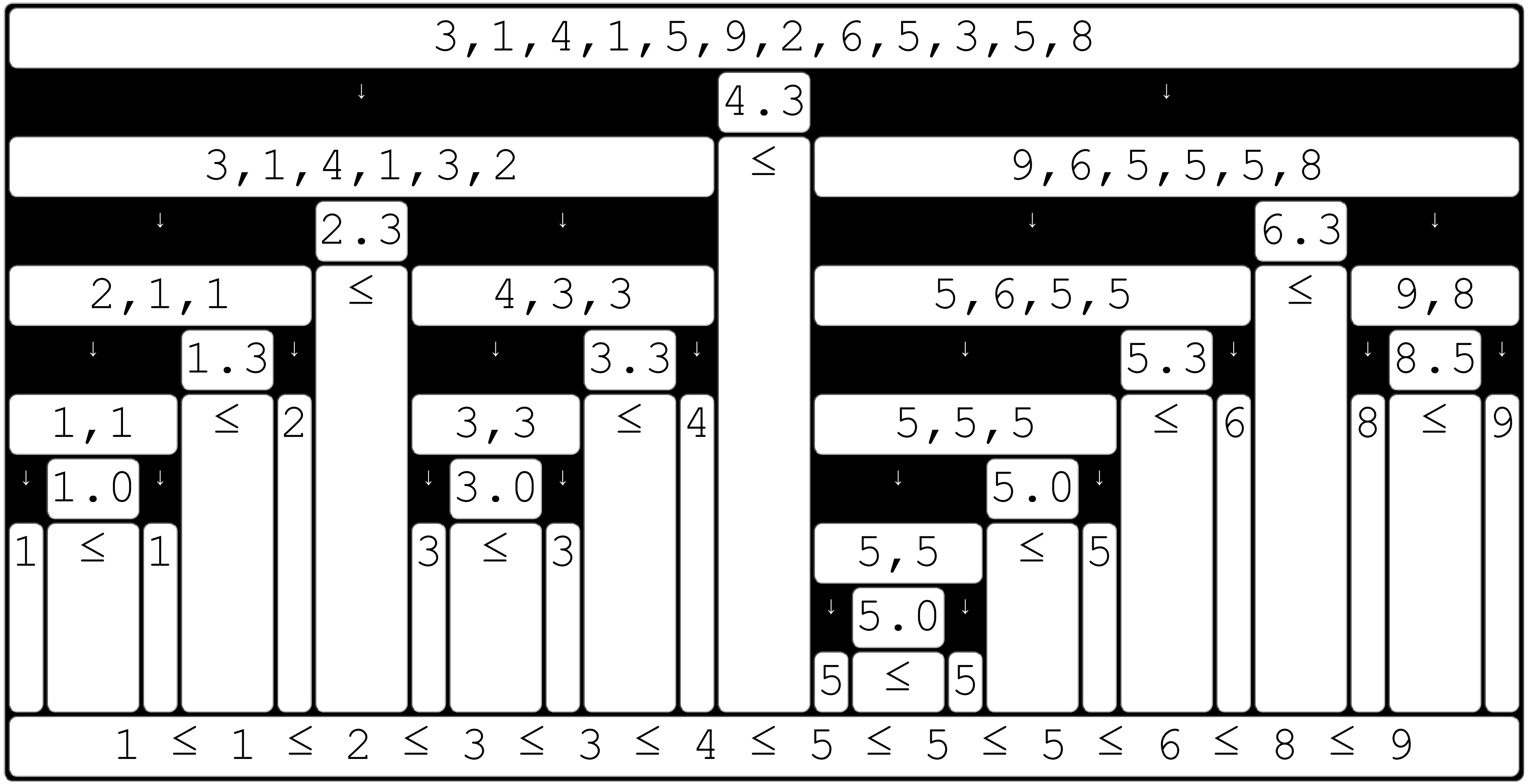
next



previous
first

As a matter of fact, Quicksort does not create any new list, it reuses the same storage space.

next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

↓ 5.0 ↓

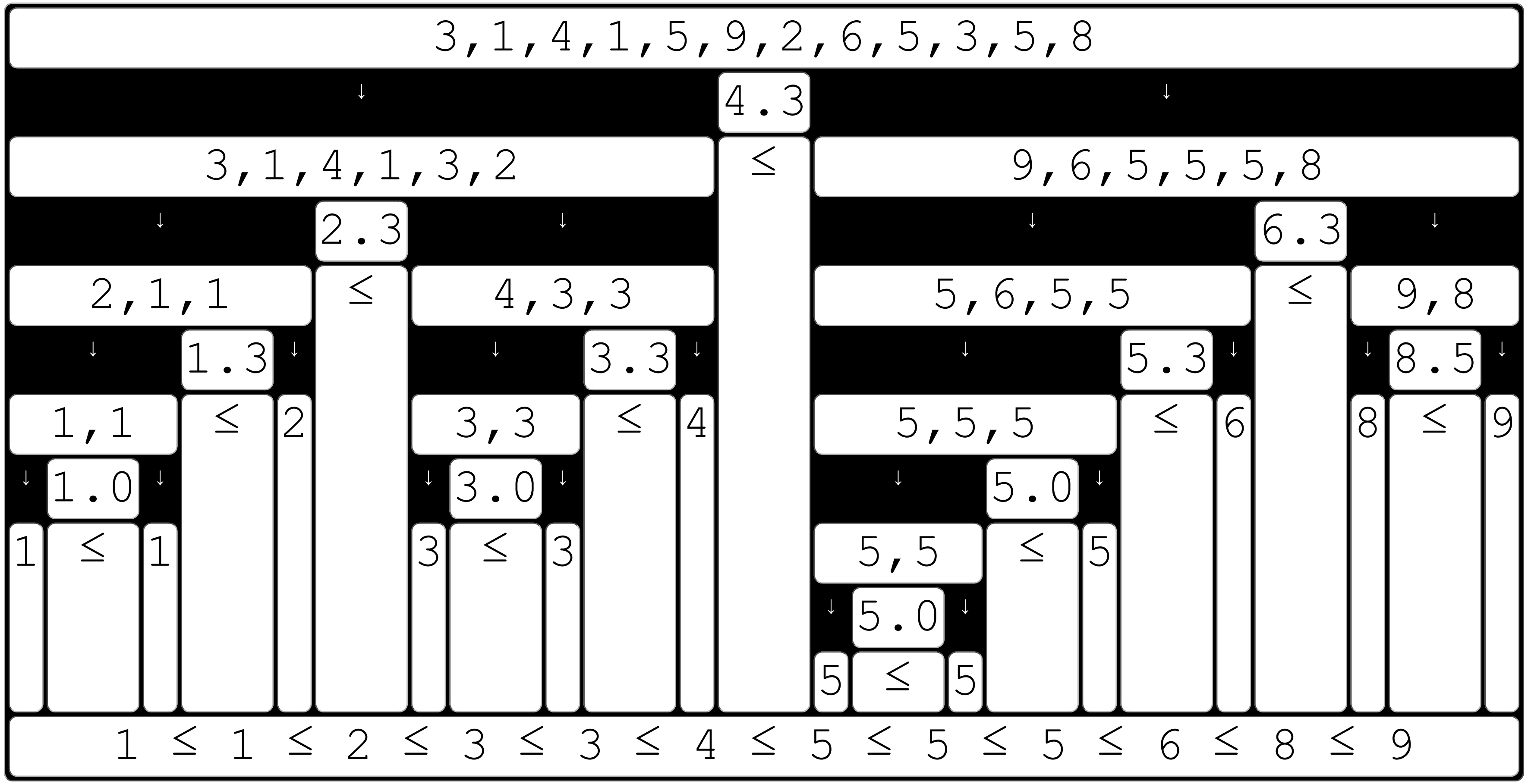
5 ≤ 5

1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

previous
first

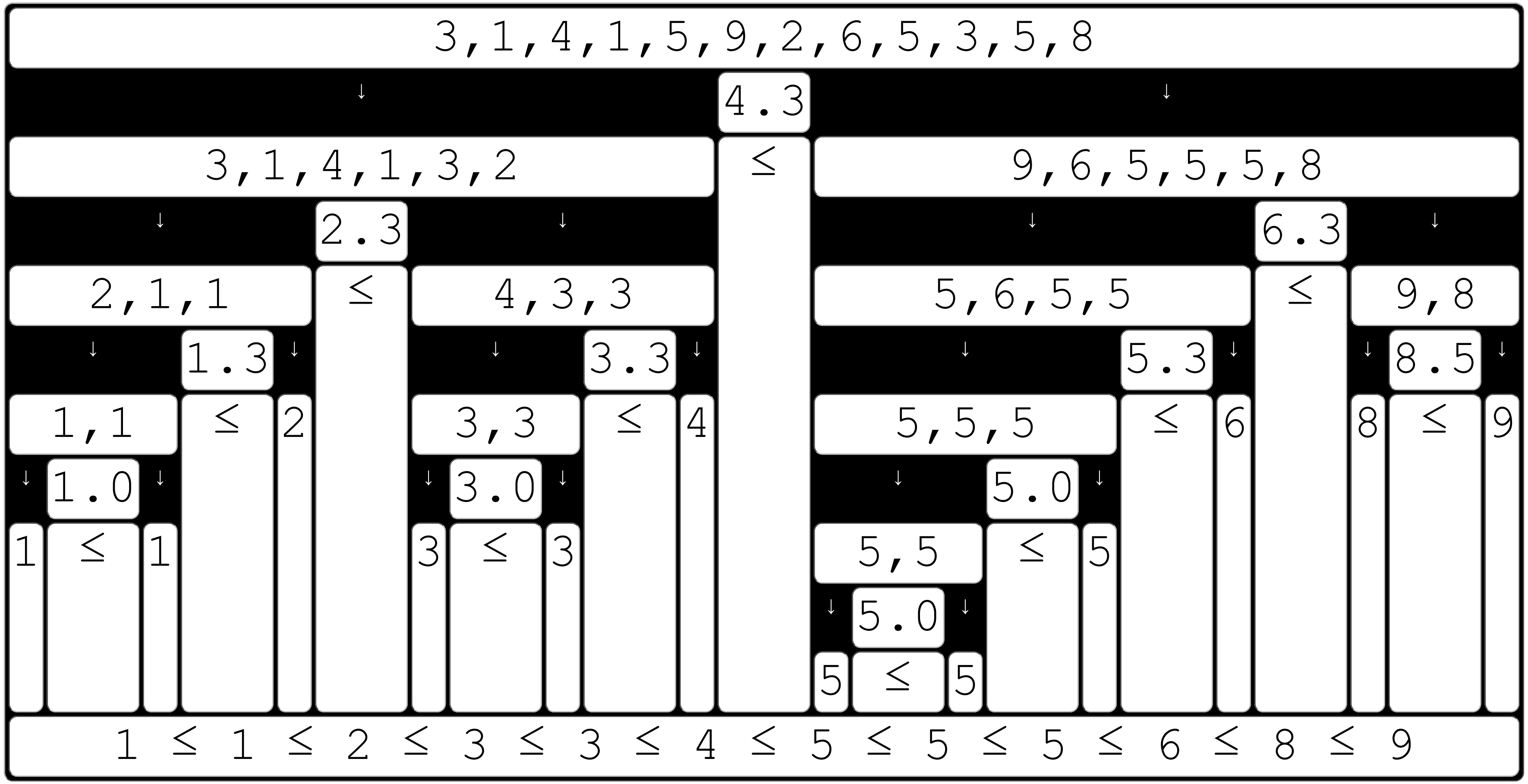
So, usually (at the split procedure) the order of equal values is not the same as the previous iteration.

next



That makes Quicksort an unstable sort algorithm.

previous
first
next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

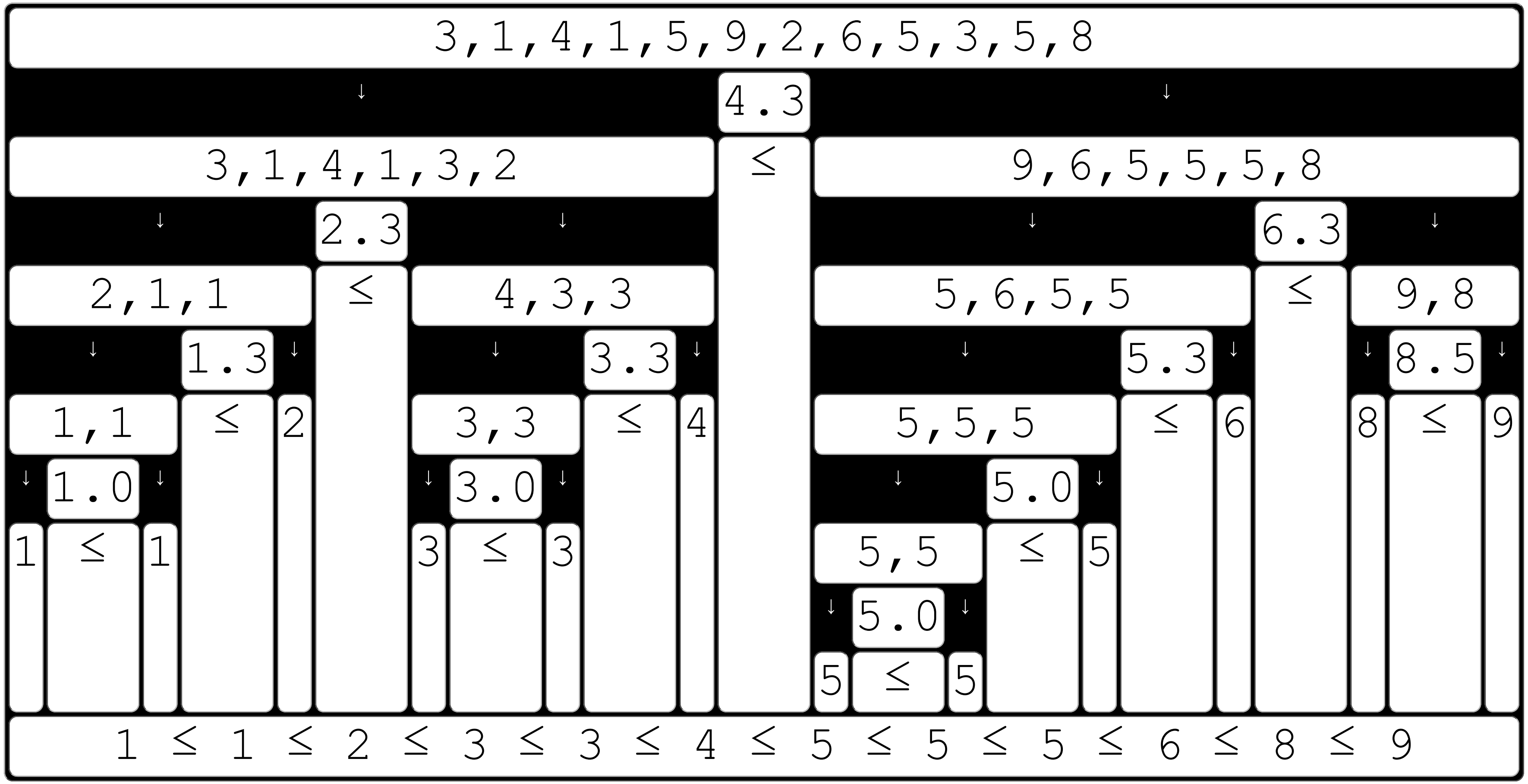
↓ 5.0 ↓

5 ≤ 5

1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

(4) Advanced concepts: stable / unstable sort algorithm.

previous
first
next

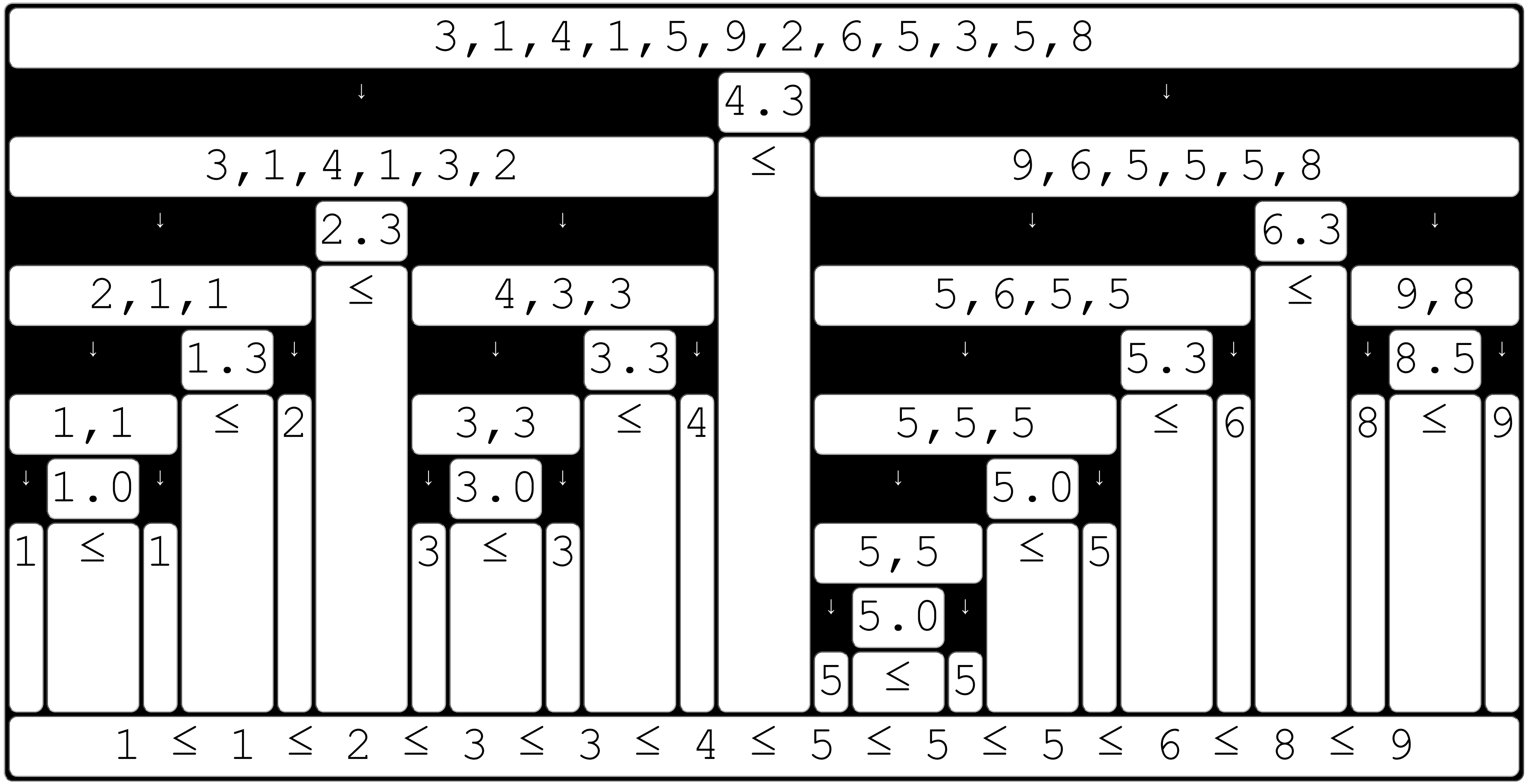


previous

first

When a list contains two or more values that are exactly the same, how do you compare those values?

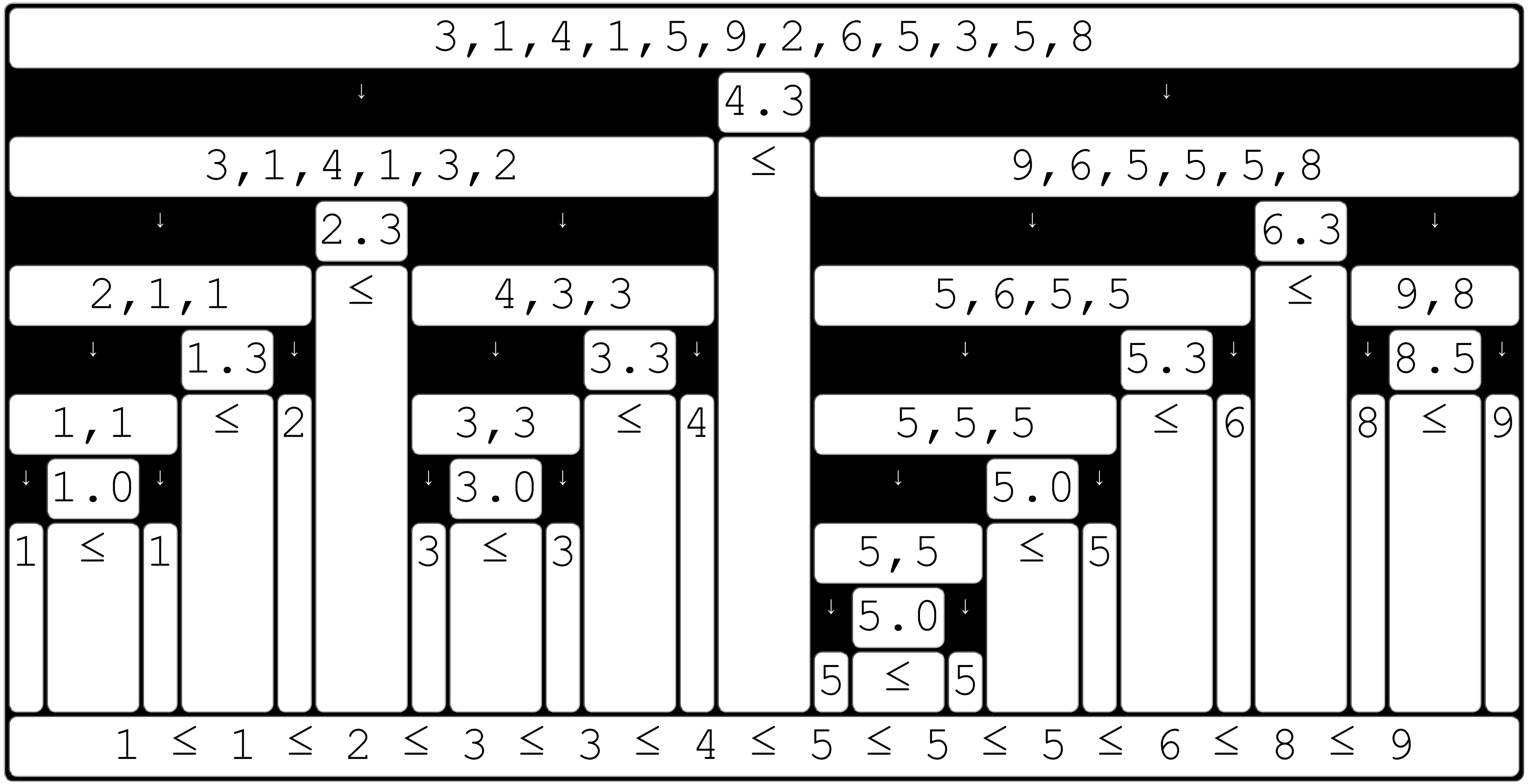
next



previous
first

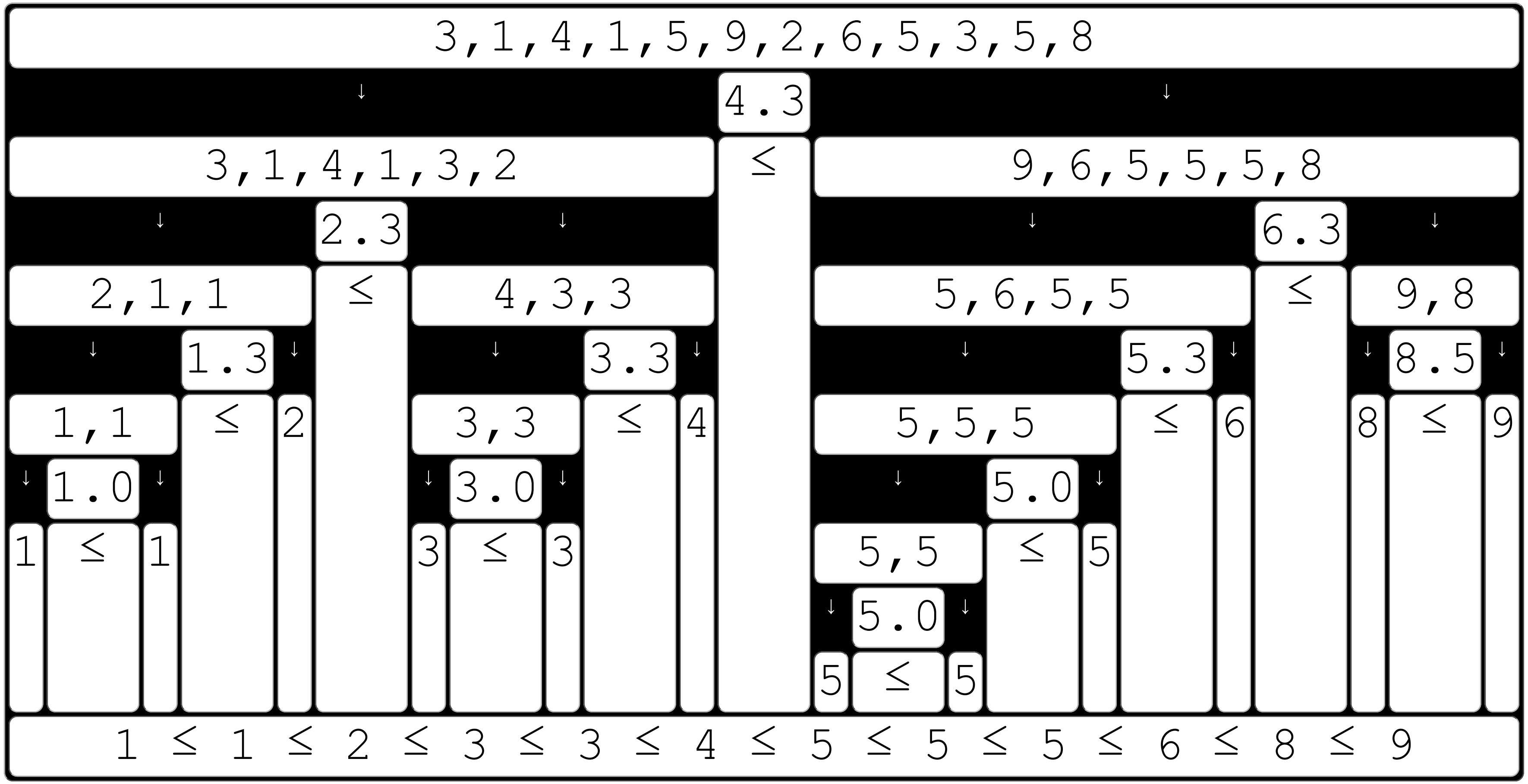
In some applications, data with the same value may use additional information (what results in equal values not being the same).

next



So, a stable / unstable sort algorithm may be relevant, for some applications.

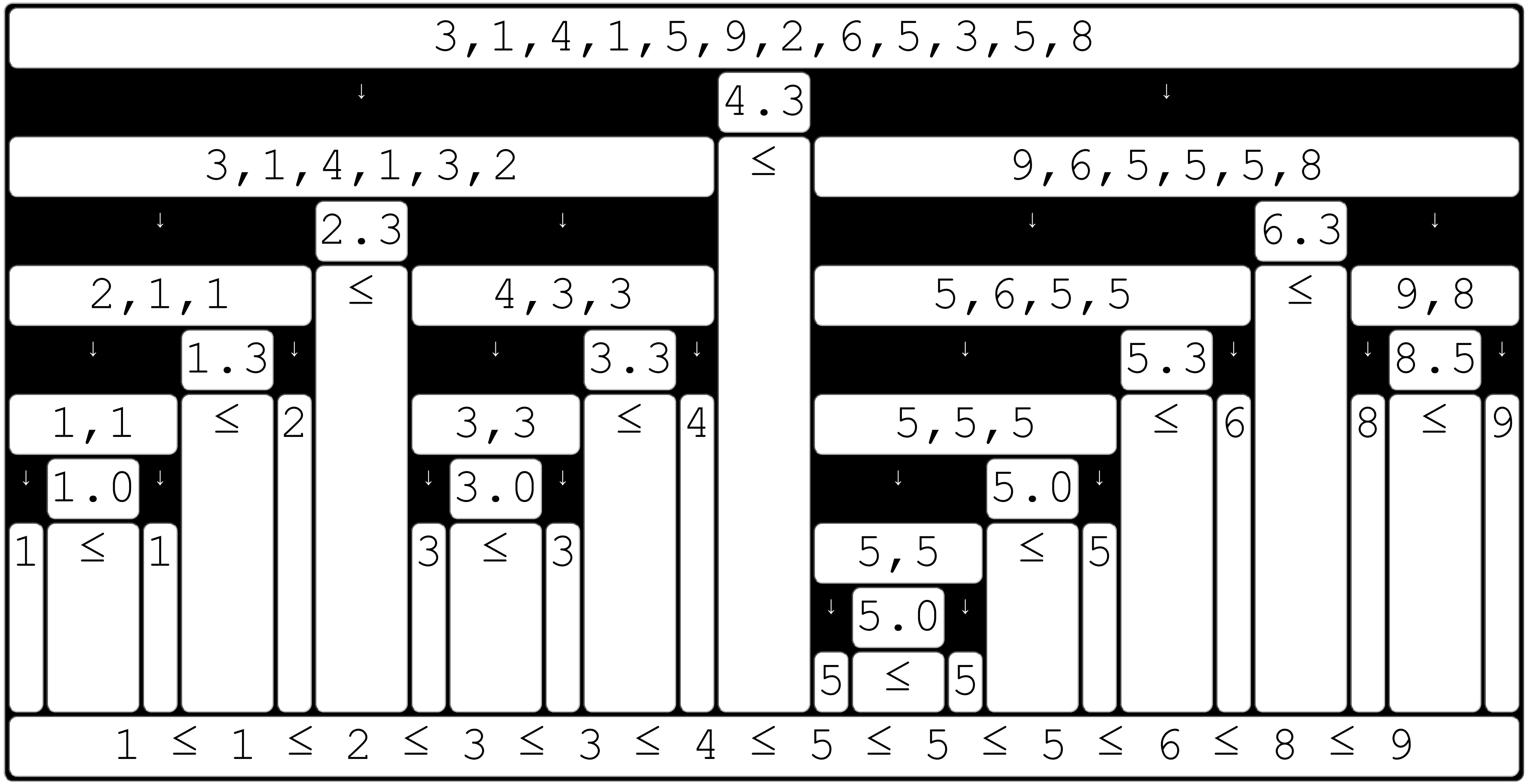
previous
first
next



previous
first

Quicksort can be implemented as a stable sort algorithm, but with some additional storage space.

next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

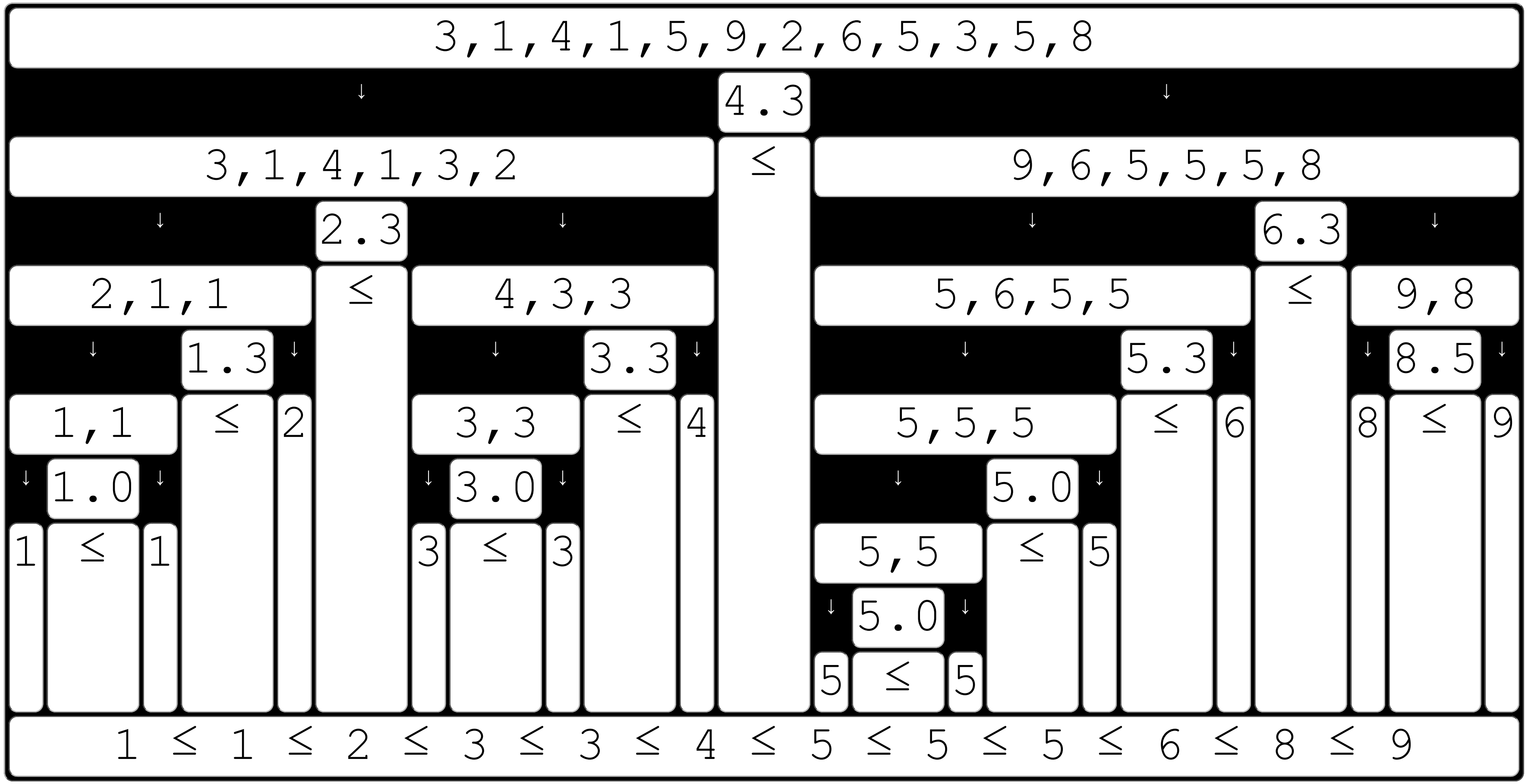
↓ 5.0 ↓

5 ≤ 5

1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

(5) Advanced concepts: Median.

previous
first
next



3,1,4,1,5,9,2,6,5,3,5,8

4.3

3,1,4,1,3,2

9,6,5,5,5,8

2.3

6.3

2,1,1

4,3,3

5,6,5,5

9,8

1.3

3.3

5.3

8.5

1,1

2

3,3

4

5,5,5

6

8

9

1.0

3.0

5.0

1

1

3

3

5,5

5

5.0

5

5

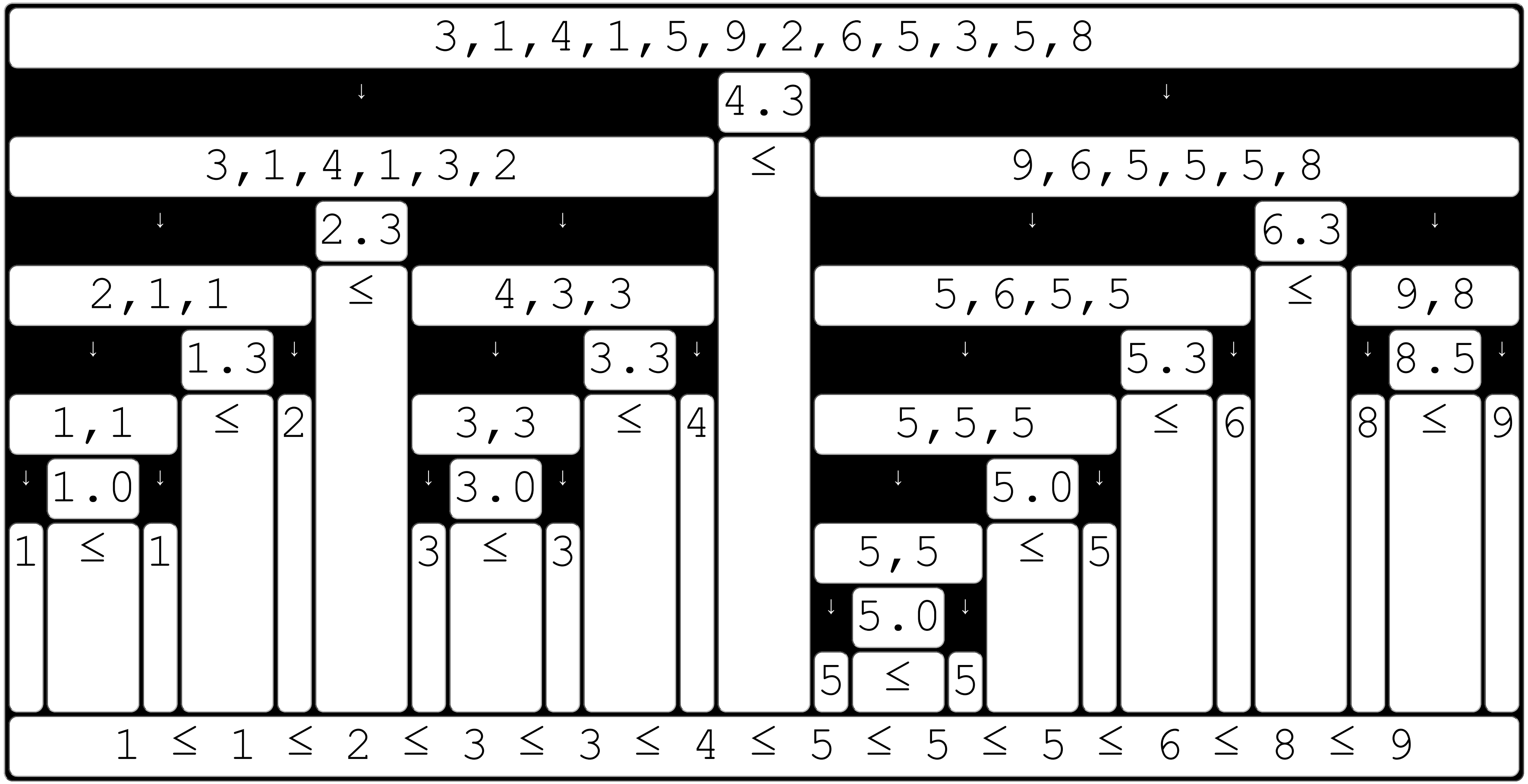
1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

previous

first

This tutorial uses average as pivot. But the pivot could have another definition.

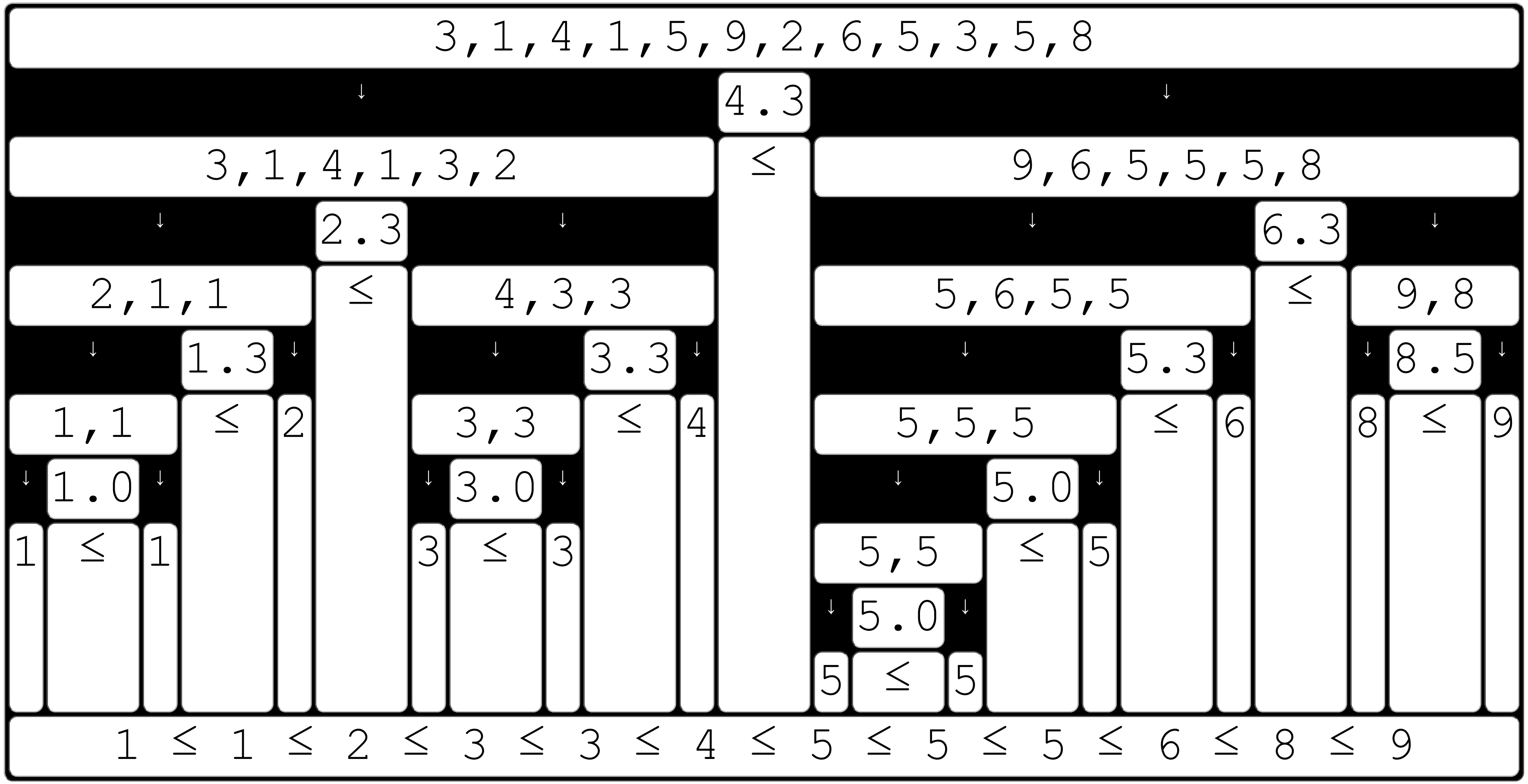
next



previous
first

The choice of pivot defines the size of each sublist. This can result in undesired unbalanced lists.

next

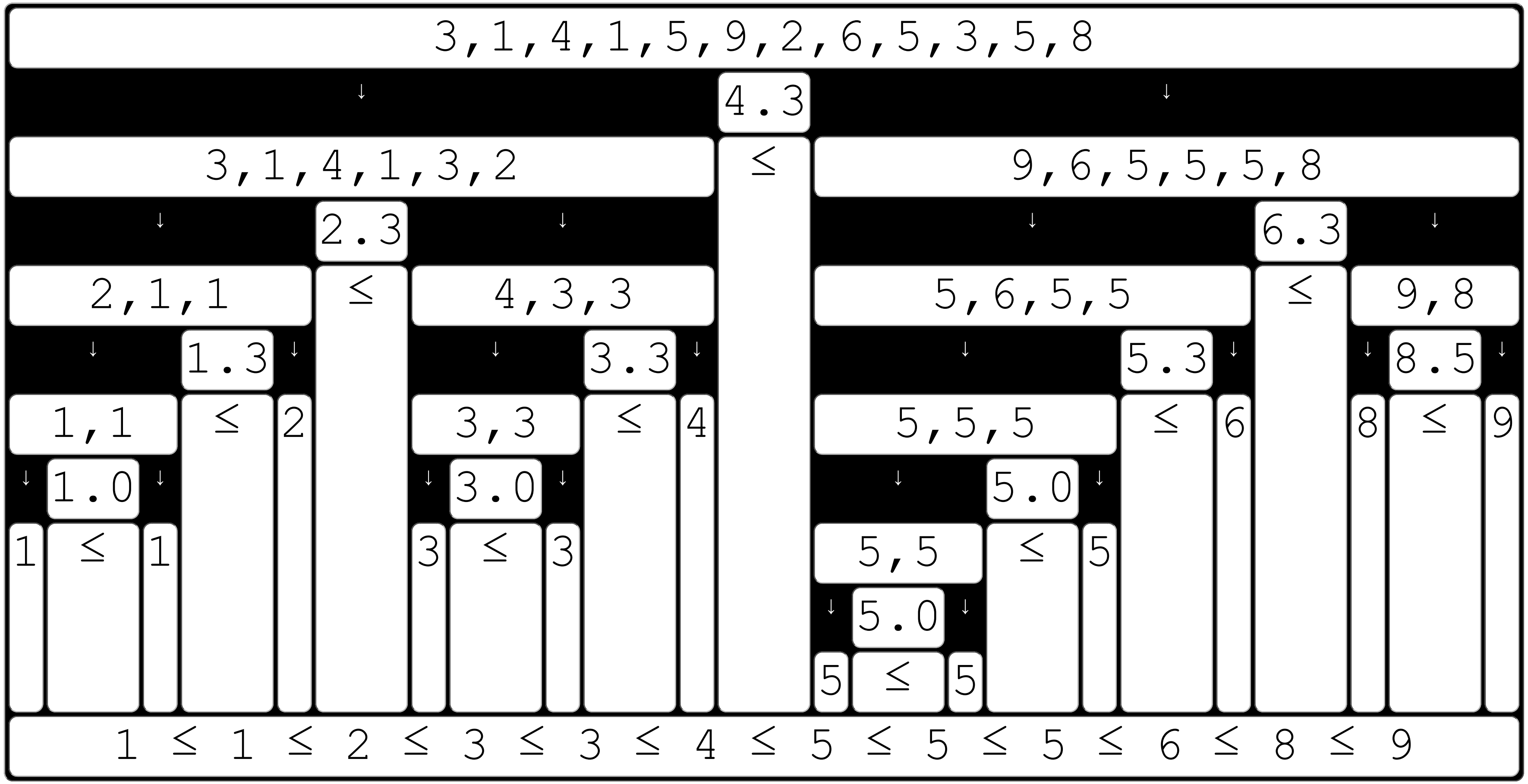


previous

first

The perfect pivot would be the median of a list.

next



3,1,4,1,5,9,2,6,5,3,5,8

↓ 4.3 ↓

3,1,4,1,3,2 ≤ 9,6,5,5,5,8

↓ 2.3 ↓ ↓ 6.3 ↓

2,1,1 ≤ 4,3,3 5,6,5,5 ≤ 9,8

↓ 1.3 ↓ ↓ 3.3 ↓ ↓ 5.3 ↓ ↓ 8.5 ↓

1,1 ≤ 2 3,3 ≤ 4 5,5,5 ≤ 6 8 ≤ 9

↓ 1.0 ↓ ↓ 3.0 ↓ ↓ 5.0 ↓

1 ≤ 1 3 ≤ 3 5,5 ≤ 5

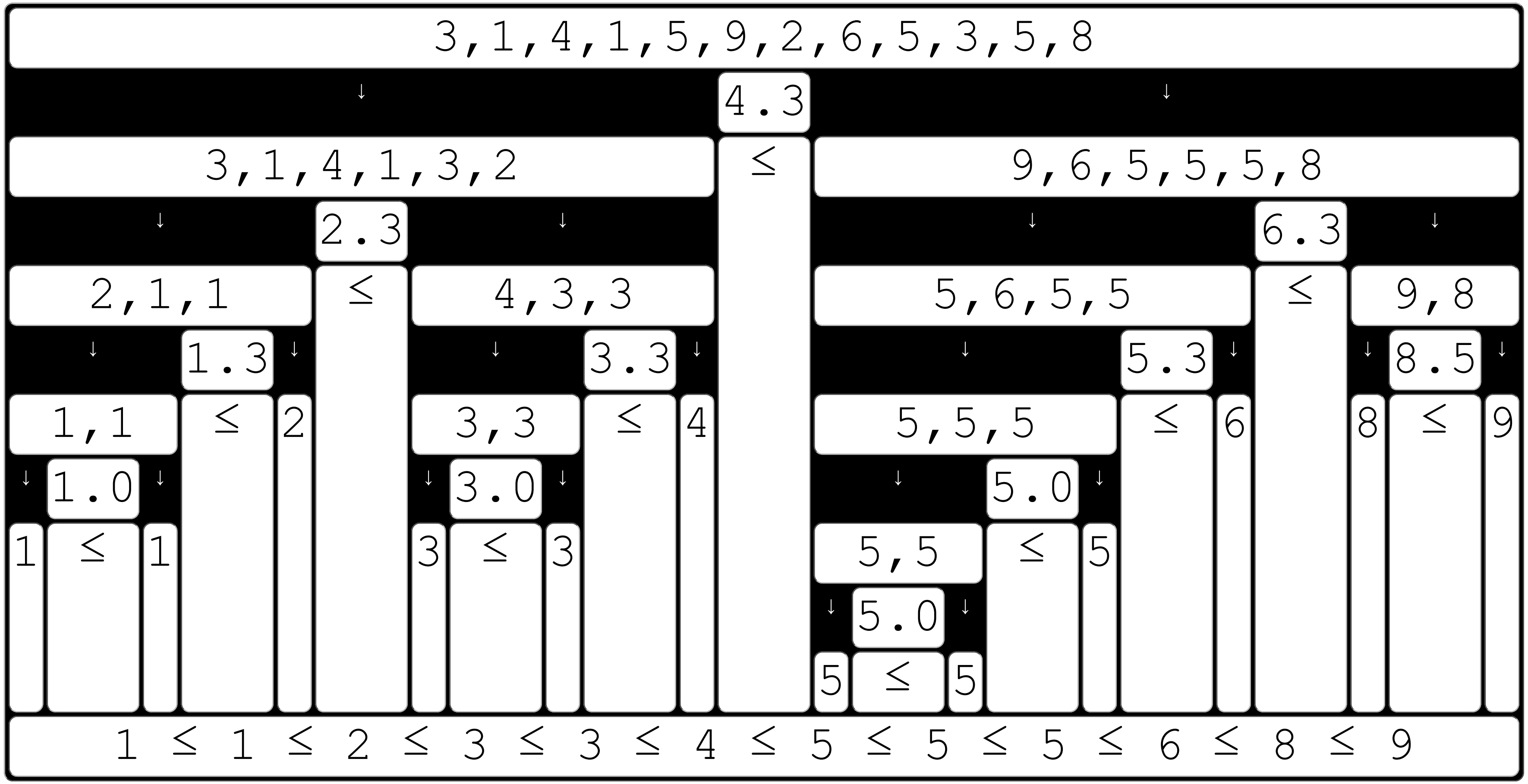
↓ 5.0 ↓

5 ≤ 5

1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

But the effort to calculate the median would make Quicksort slower.

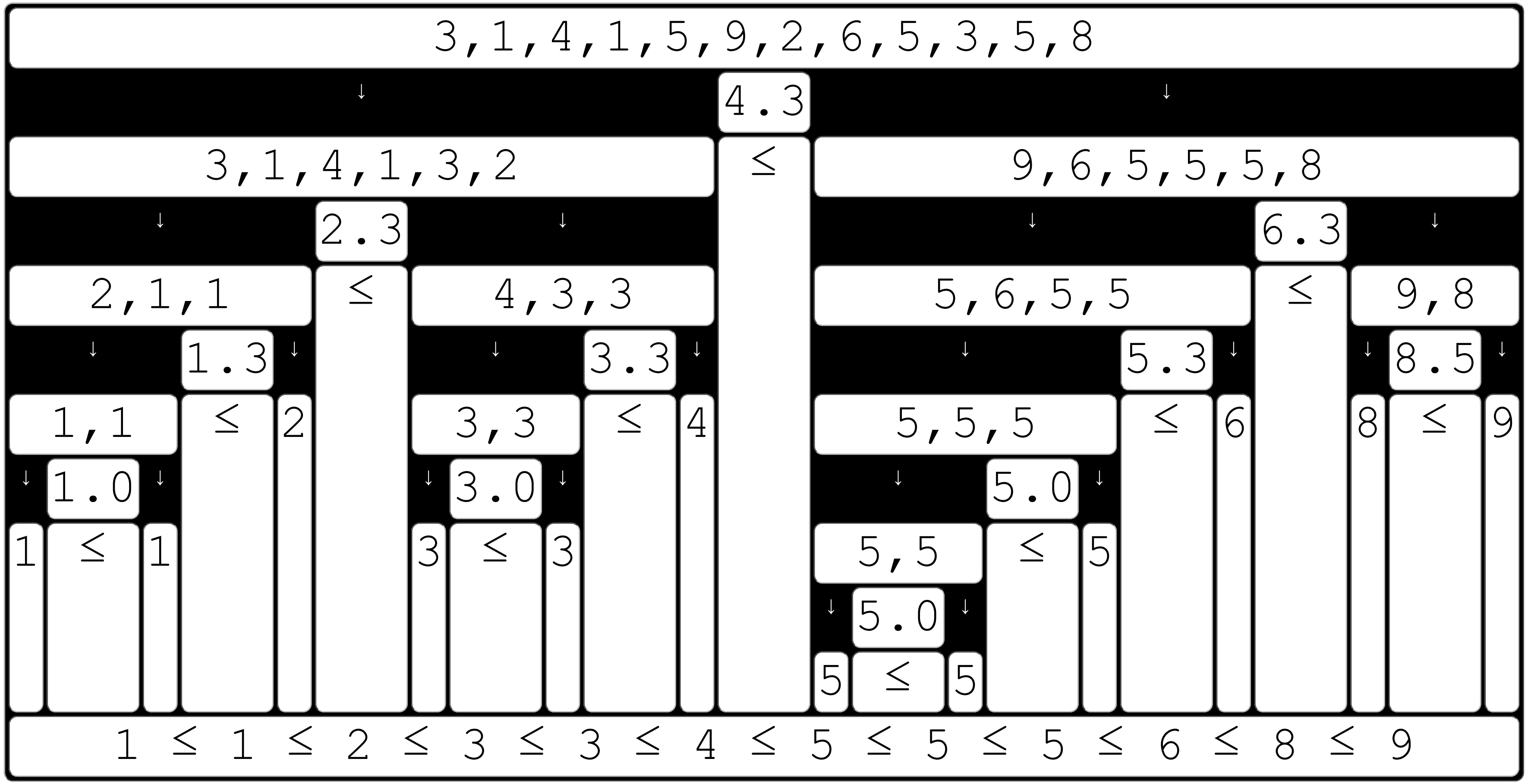
previous
first
next



previous
first

So, in practice, some other value is defined as pivot: as near to the median as possible, and fast to calculate.

next



3,1,4,1,5,9,2,6,5,3,5,8

4.3

3,1,4,1,3,2

≤

9,6,5,5,5,8

2.3

6.3

2,1,1

4,3,3

5,6,5,5

9,8

1.3

3.3

5.3

8.5

1,1

2

3,3

4

5,5,5

6

8

9

1.0

3.0

5.0

1 ≤ 1

3 ≤ 3

5,5

5

5.0

5 ≤ 5

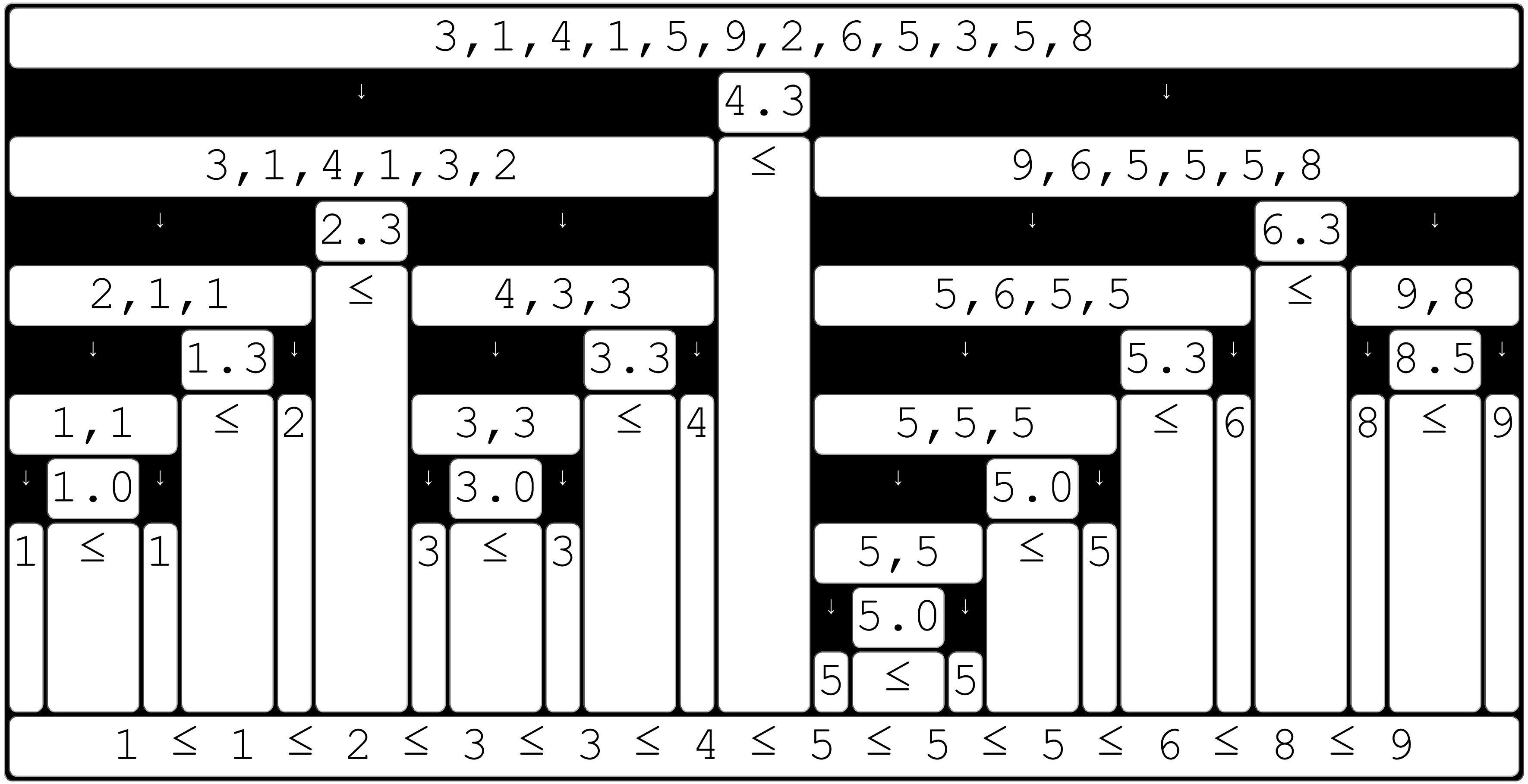
1 ≤ 1 ≤ 2 ≤ 3 ≤ 3 ≤ 4 ≤ 5 ≤ 5 ≤ 5 ≤ 6 ≤ 8 ≤ 9

previous

first

Originally in 1959, Tony Hoare implemented Quicksort using the first (or last) element of a list as pivot. The algorithm usually works fine with this choice.

next



previous

first

The end.
Also, visit Mergesort Interactive Tutorial.

next

Quicksort
Tutorial Highlight

https://www.ime.usp.br/~otuyama
/tutorial/sort/quicksort/Quicksort.html