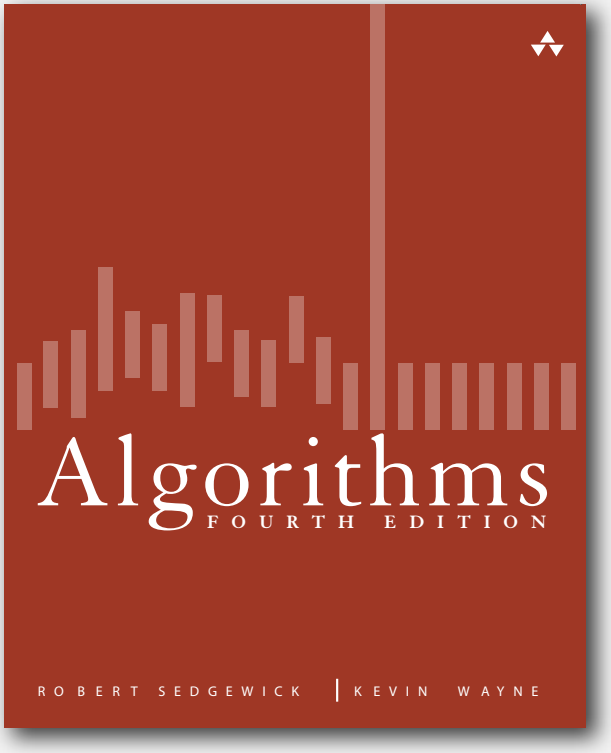
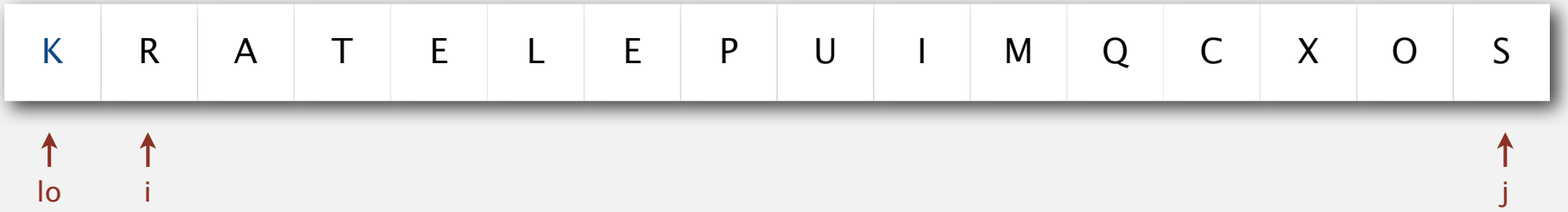
Sedgewick 2-way Partitioning



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

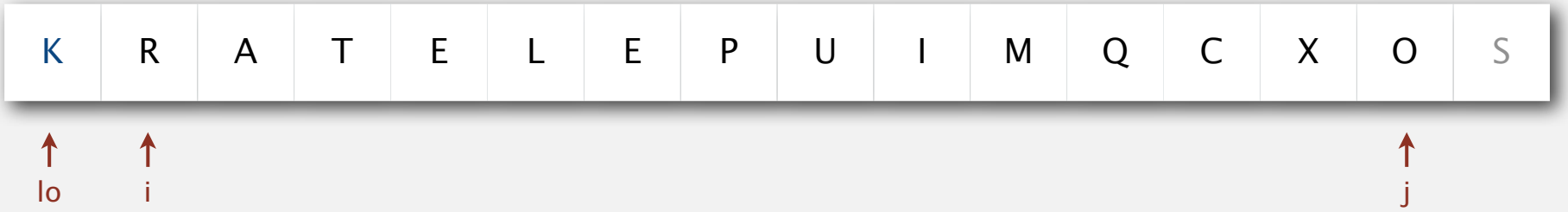


stop `i` scan because `a[i] >= a[lo]`

Quicksort partitioning

Repeat until `i` and `j` pointers cross.

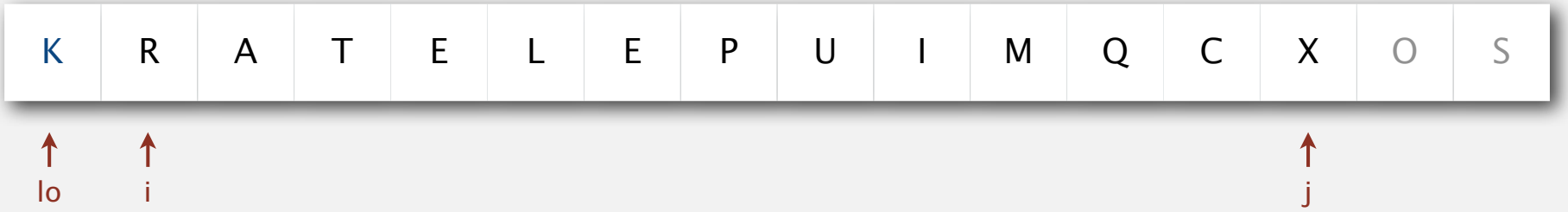
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

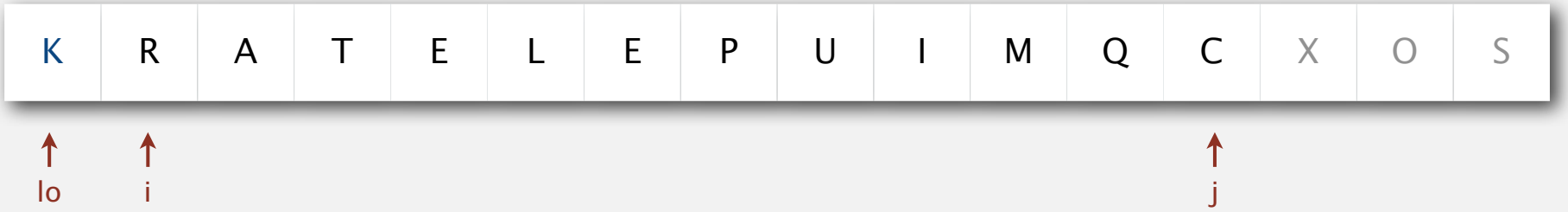
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

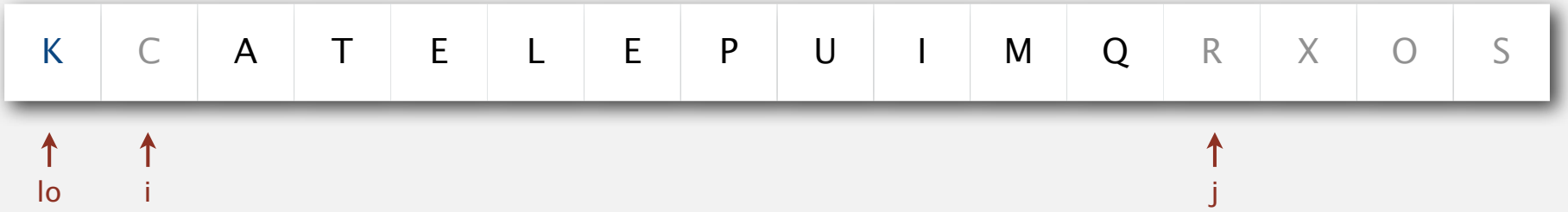


stop `j` scan and exchange `a[i]` with `a[j]`

Quicksort partitioning

Repeat until `i` and `j` pointers cross.

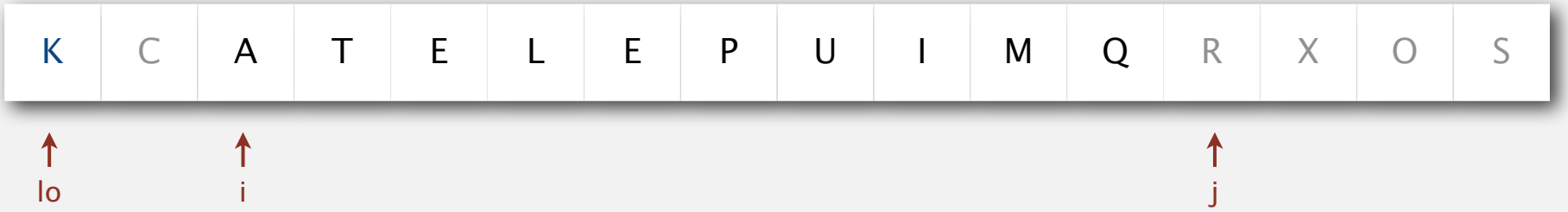
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

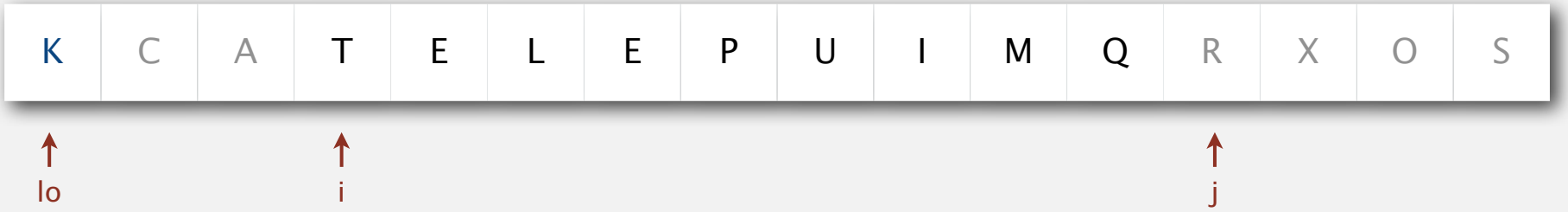
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

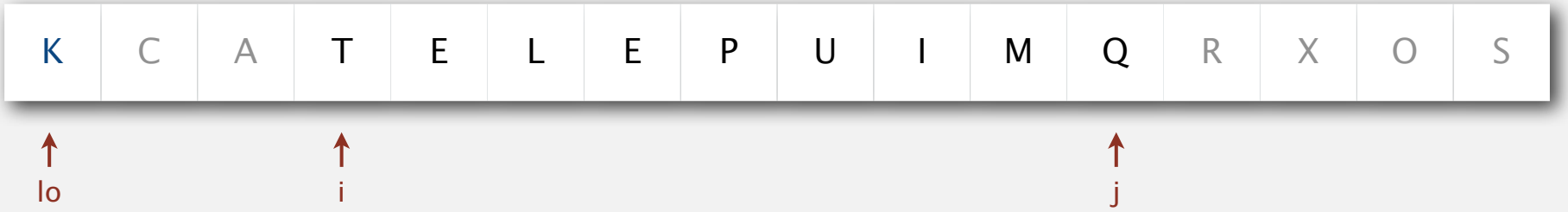


stop `i` scan because `a[i] >= a[lo]`

Quicksort partitioning

Repeat until `i` and `j` pointers cross.

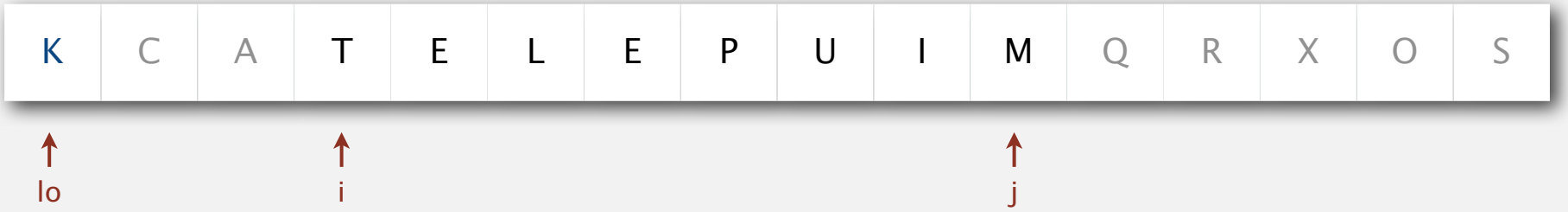
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

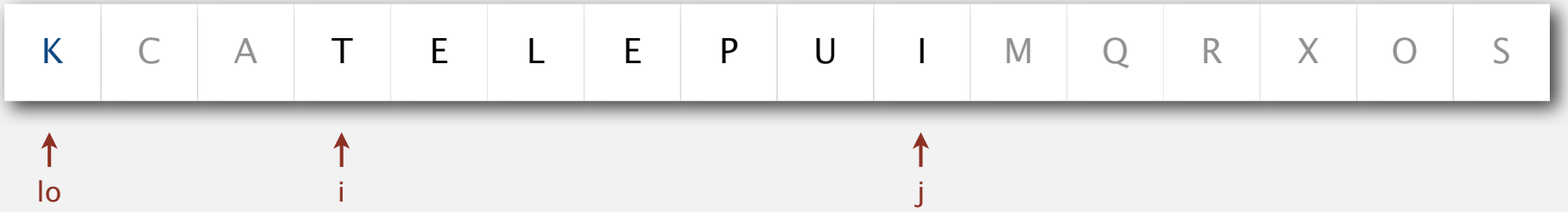
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

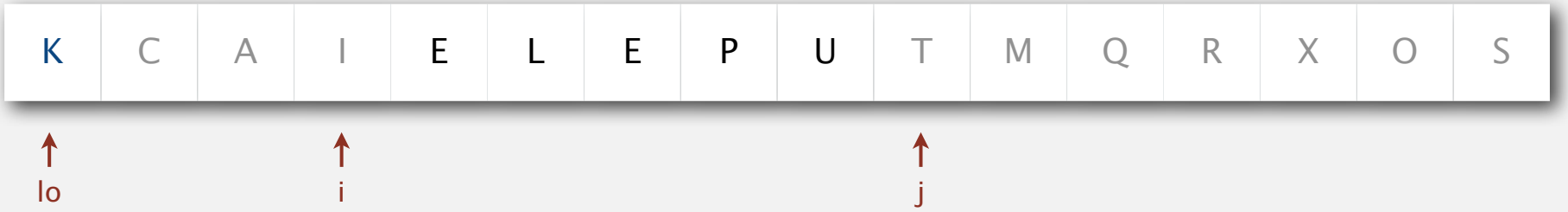


stop `j` scan and exchange `a[i]` with `a[j]`

Quicksort partitioning

Repeat until `i` and `j` pointers cross.

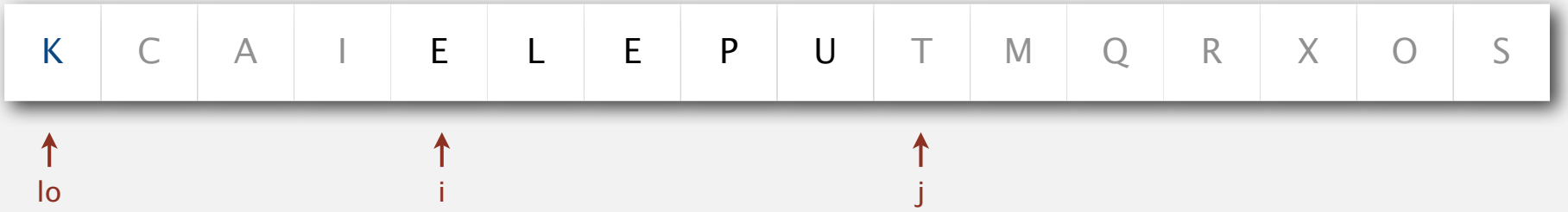
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

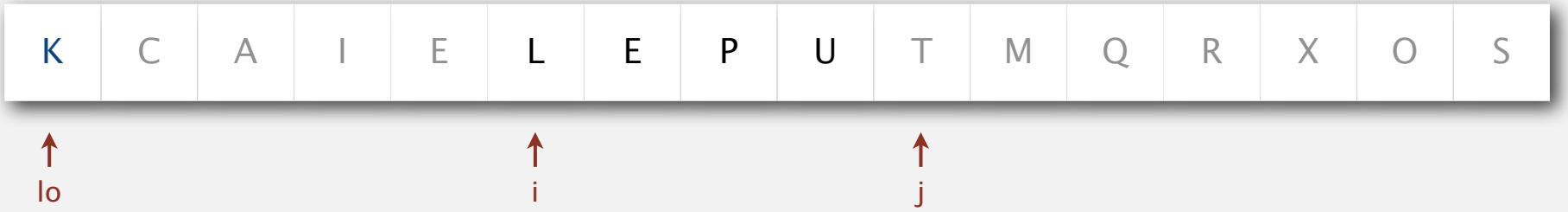
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

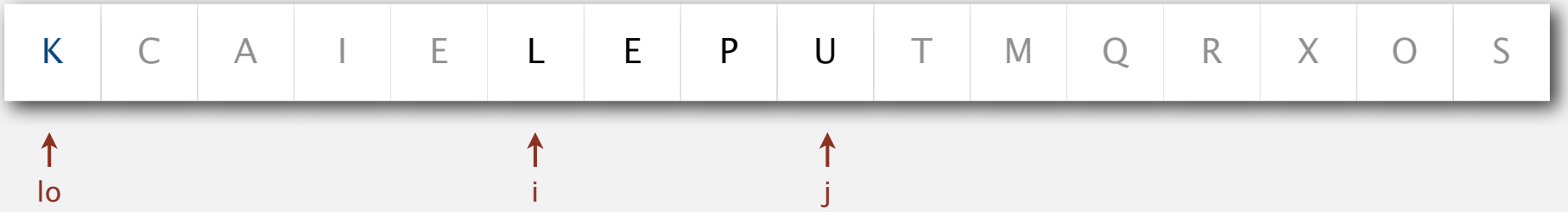


stop `i` scan because `a[i] >= a[lo]`

Quicksort partitioning

Repeat until `i` and `j` pointers cross.

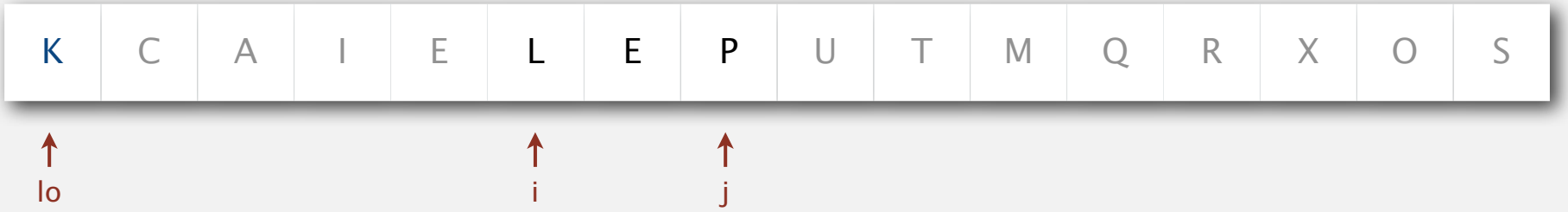
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

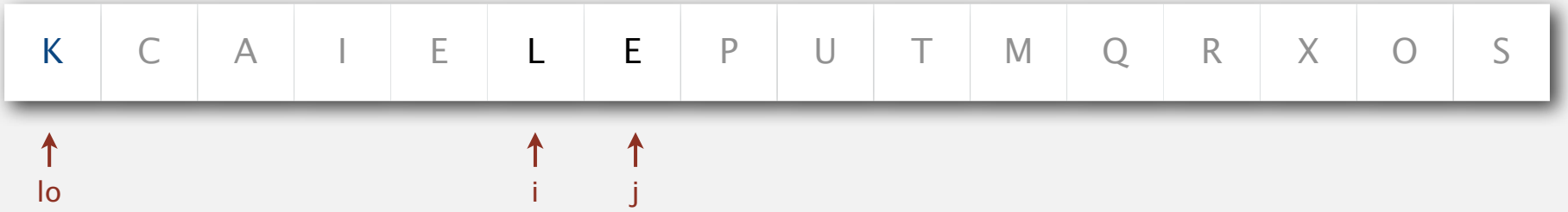
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

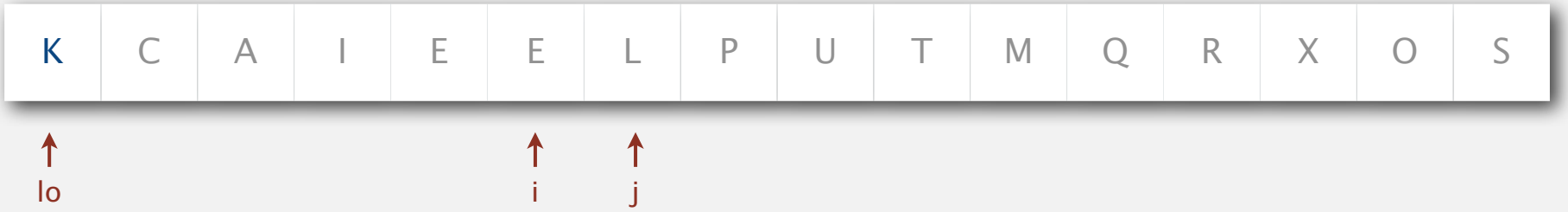


stop `j` scan and exchange `a[i]` with `a[j]`

Quicksort partitioning

Repeat until `i` and `j` pointers cross.

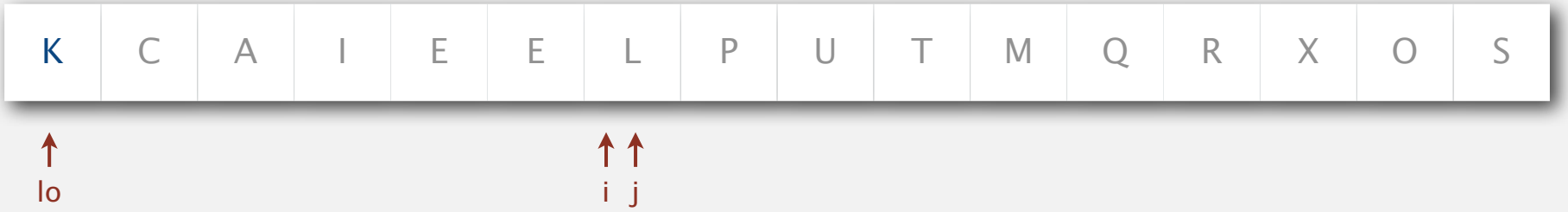
- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

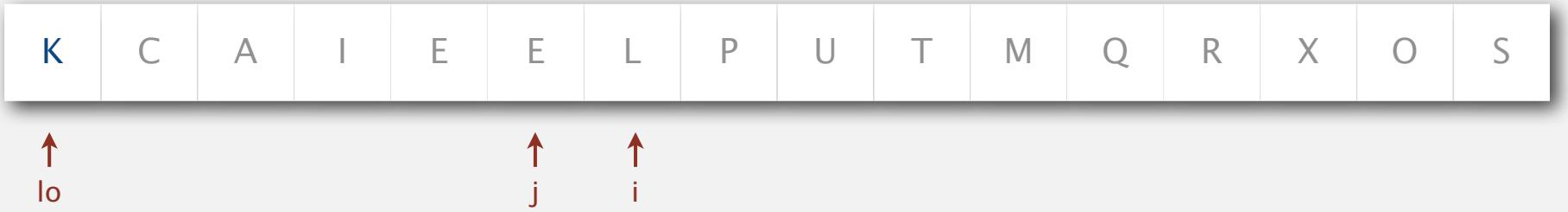


stop `i` scan because `a[i] >= a[lo]`

Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.



stop `j` scan because `a[j] <= a[lo]`

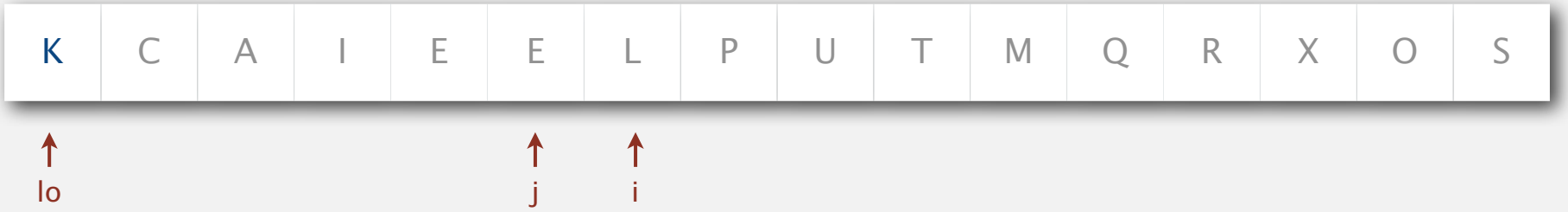
Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

When pointers cross.

- Exchange `a[lo]` with `a[j]`.



pointers cross: exchange `a[lo]` with `a[j]`

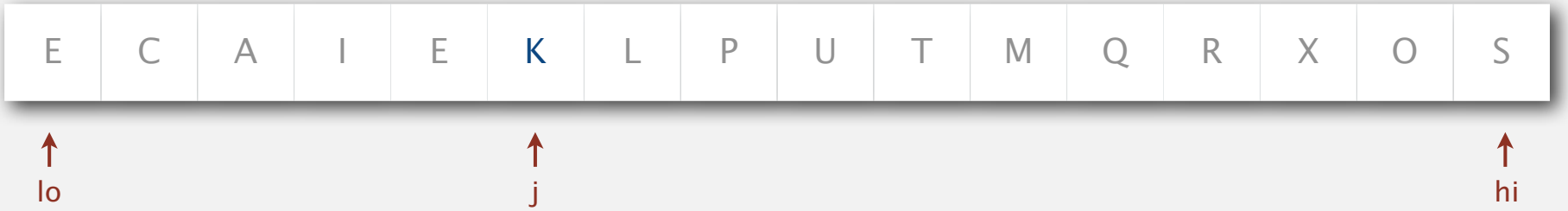
Quicksort partitioning

Repeat until `i` and `j` pointers cross.

- Scan `i` from left to right so long as `a[i] < a[lo]`.
- Scan `j` from right to left so long as `a[j] > a[lo]`.
- Exchange `a[i]` with `a[j]`.

When pointers cross.

- Exchange `a[lo]` with `a[j]`.



partitioned!