

Algorithm Analysis: Searching

Fall 2013

Carola Wenk

Searching A List

A very common task is to check whether an item is contained in a list.

L:

| | | | | | | | | | |
|---|---|---|---|----|-----|---|---|----|---|
| 2 | 5 | 1 | 0 | -1 | ... | 8 | 6 | -3 | 4 |
|---|---|---|---|----|-----|---|---|----|---|

Does L contain the number 6?

1. Define the problem (input, output)

Input

A list L and an item x to check whether it is contained in L.

Output

The item's index, if it is in the list.

What should the output be if x is not in L?

Searching A List

A very common task is to check whether an item is contained in a list.

L:

| | | | | | | | | | |
|---|---|---|---|----|-----|---|---|----|---|
| 2 | 5 | 1 | 0 | -1 | ... | 8 | 6 | -3 | 4 |
|---|---|---|---|----|-----|---|---|----|---|

Does L contain the number 6?

2. Describe the algorithm

Scan the list L from left to right and compare it with x. If we find x, return the index. If not, return -1.

Runtime is linear in the size of the list.

⇒ **Linear search**

Sorted Lists

- Sorting is constantly used, but in many applications, the data does not change frequently and we only need to sort it once.

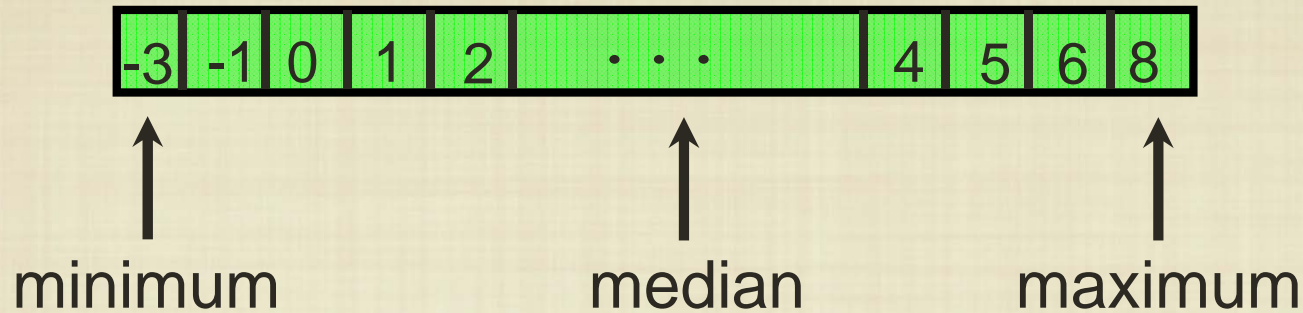


→
increasing

Where is the minimum element? The maximum element? The median?

A very common task is to check whether an item is in our (typically large) list. If the list is sorted, can we do better than searching the entire list?

Searching A Sorted List



1. Define the problem (input, output)

Input

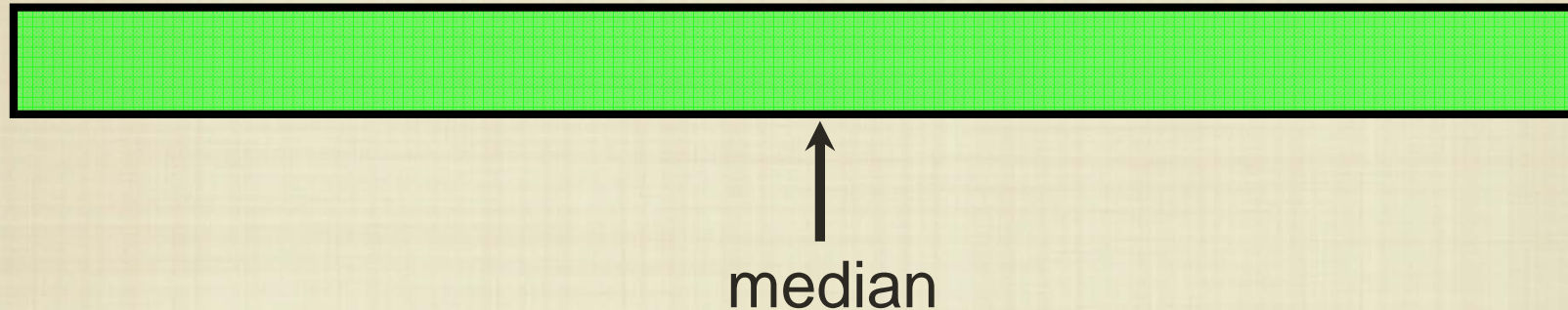
A list L sorted in increasing order, and an item x to check whether it is contained in L .

Output

The item's index, if it is in the list.

-1 if x is not in L .

Searching A Sorted List

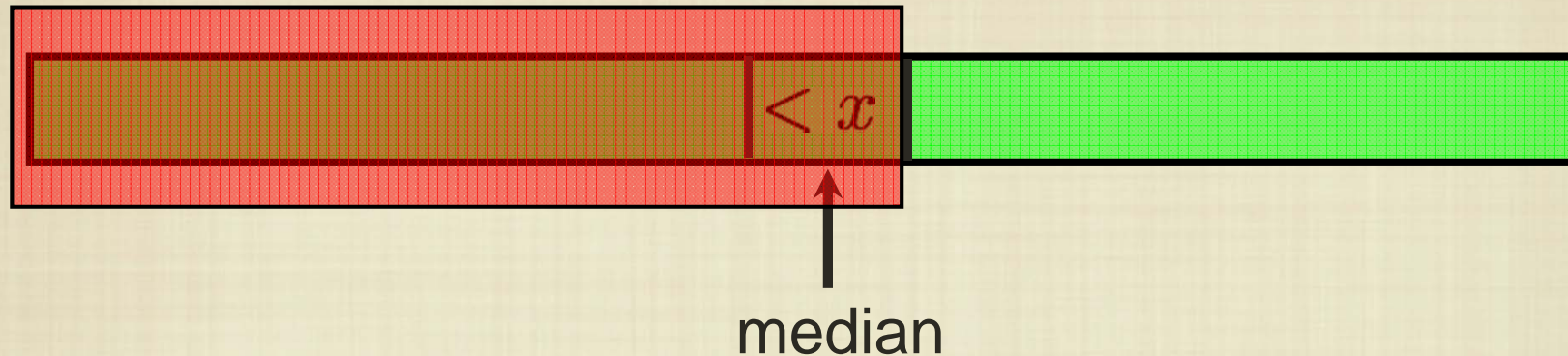


Suppose we are looking for some element, call it x .

What do we know if the median is smaller than x ? If it is larger?

By definition, half of the elements are smaller than the median and half of elements are greater. What can we do?

Searching A Sorted List



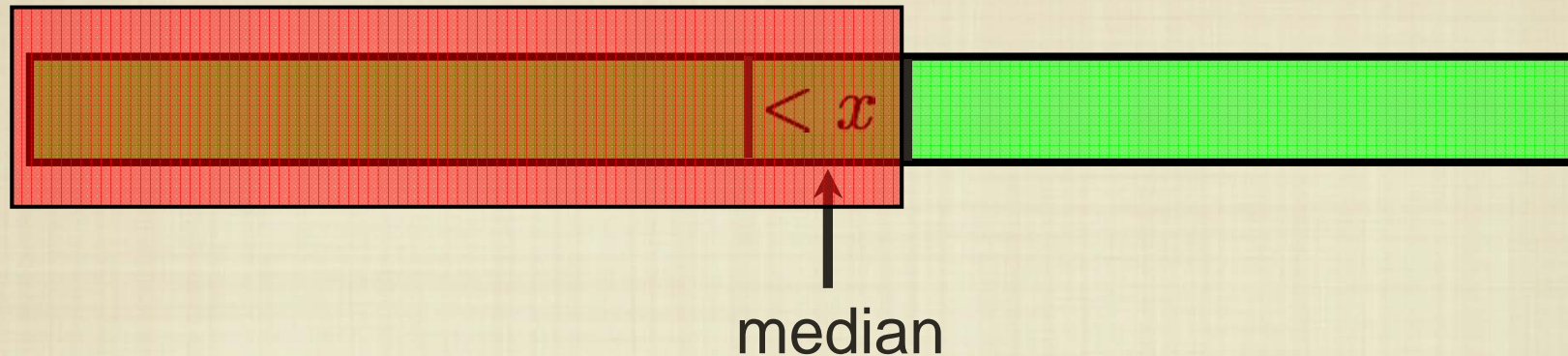
Suppose we are looking for some element, call it x .

What do we know if the median is smaller than x ? If it is larger?

By definition, half of the elements are smaller than the median and half of the elements are greater.

What can we do?

Searching A Sorted List

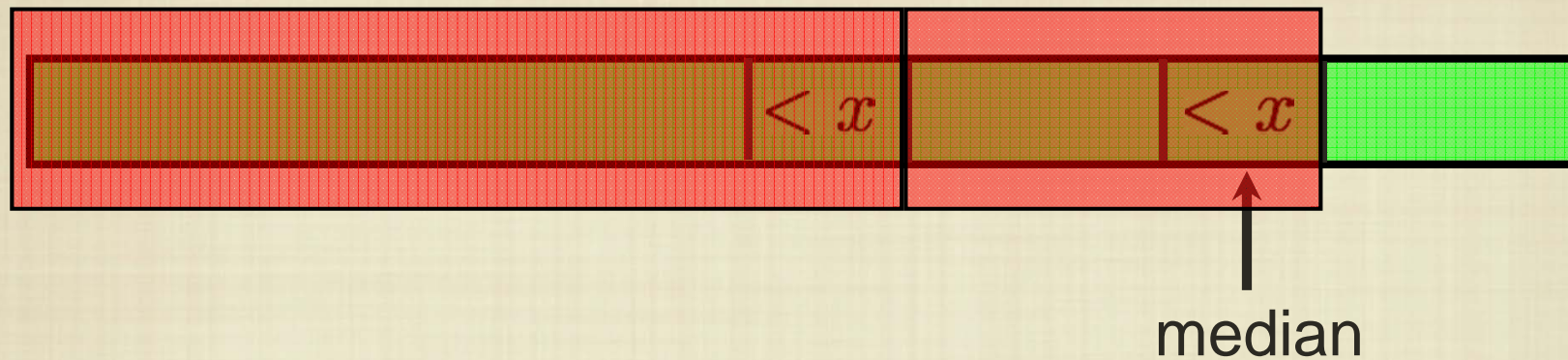


Suppose we are looking for some element, call it x .

What do we know if the median is smaller than x ? If it is larger?

By inspecting the median, we can decide which half of the list to eliminate from consideration with a single comparison.

Searching A Sorted List

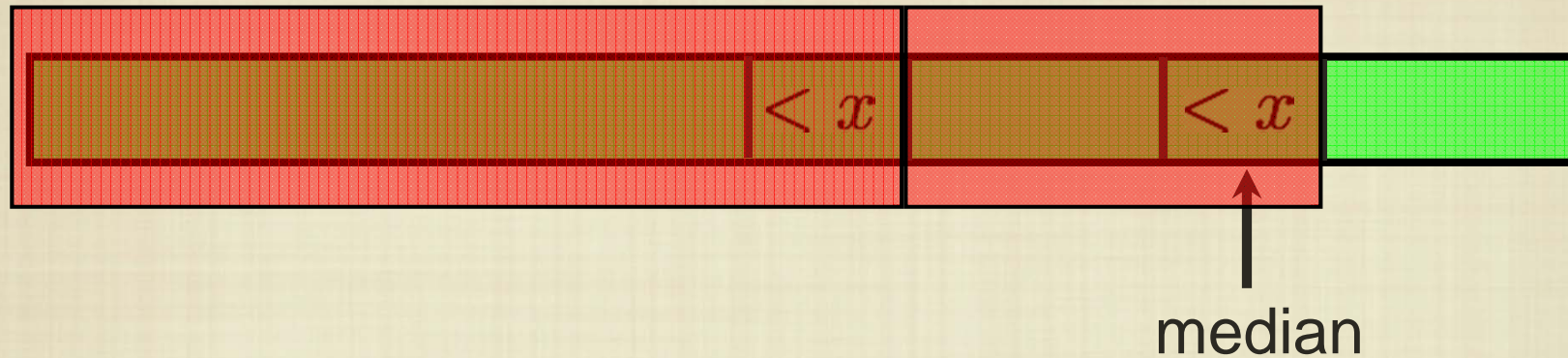


Suppose we are looking for some element, call it x .

What do we know if the median is smaller than x ? If it is larger?

By inspecting the median, we can decide which half of the list to eliminate from consideration with a single comparison.

Searching A Sorted List



Binary Search:

1. Test whether x is less than the median (if it is equal, we are done).
2. Continue to search the half of the list that x is in.
3. We are done when the “correct” side of the list is empty.