

B i n a r y S e a r c h

B i n a r y S e a r c h

A k a

H a l f - i n t e r v a l S e a r c h

L o g a r i t h m S e a r c h

B i n a r y C h o p

A l g o r i t h m : C h e c k t h e c a n d i d a t e a t t h e m

O n e o f t h e m o s t i m p o r t a n t s e a r c h i n g a l g



B i n a r y S e a r c h

Example: Find the largest number less than

Requirement: The array must be sorted



Binary Search

Requirement: The array must be sorted

Algorithm: Check the element at the middle of the first-half, all other elements are second-half

If it is smaller than or equal to 7, return its index

Otherwise (greater than 7), discard the first-half

Until there is only one element left, check it

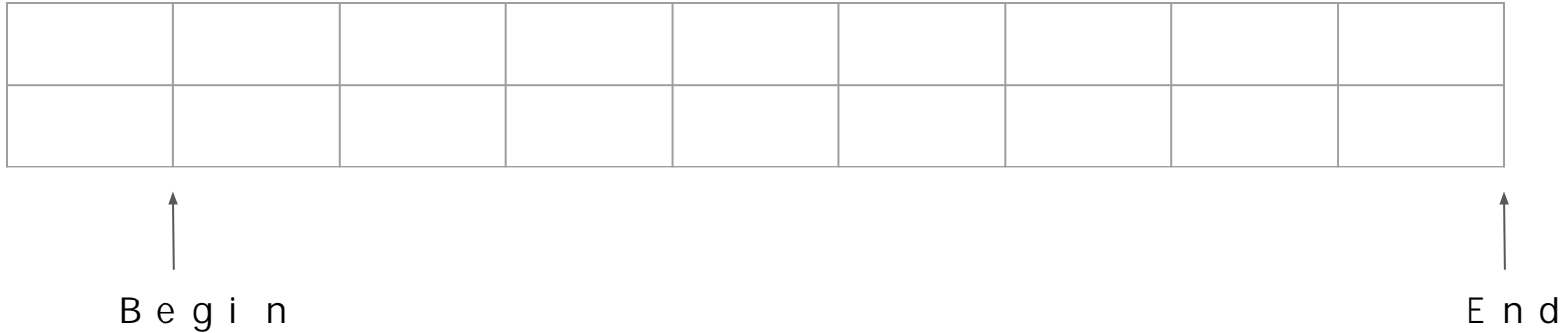
If yes, then it is the answer.

If no, then there is no answer.



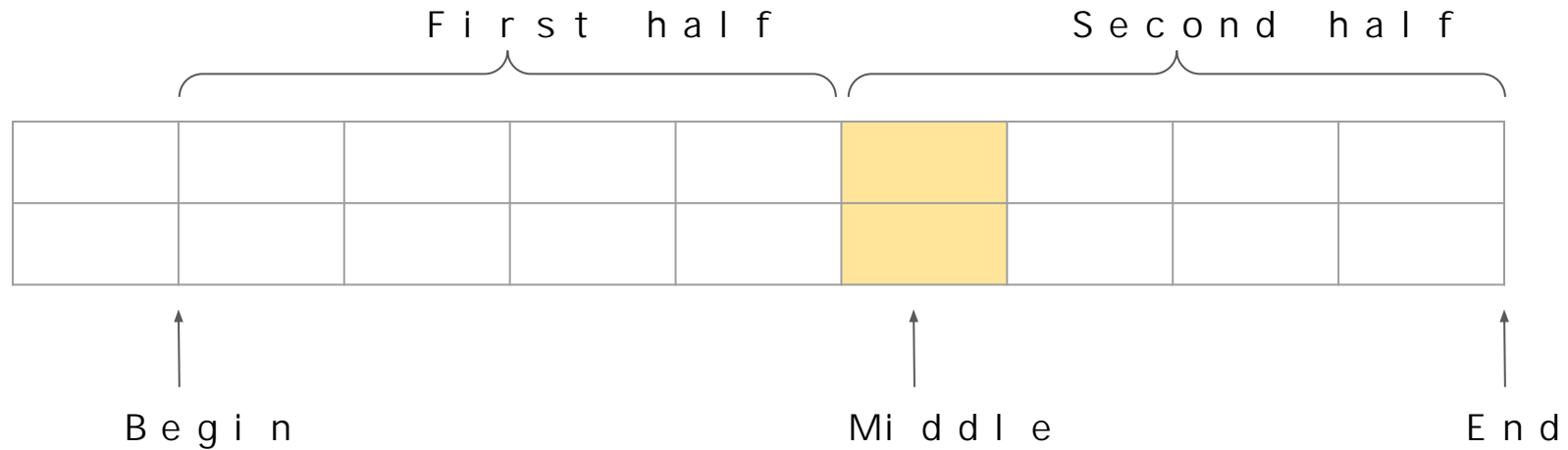
Binary Search

Example: Find the largest number less than



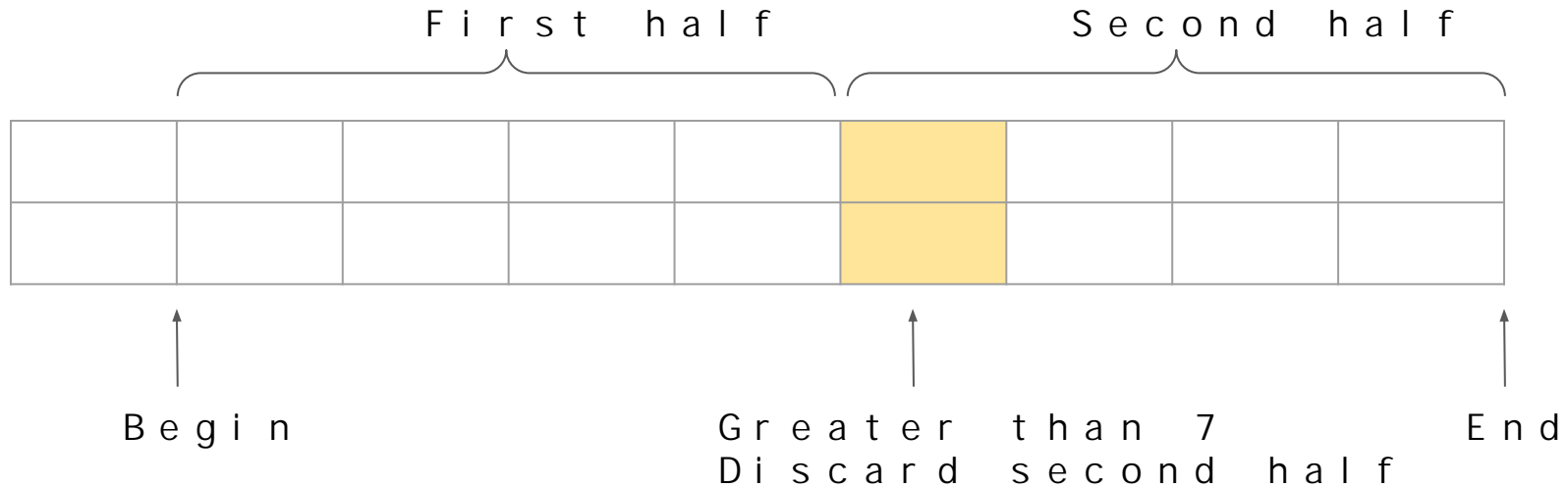
Binary Search

Example: Find the largest number less than



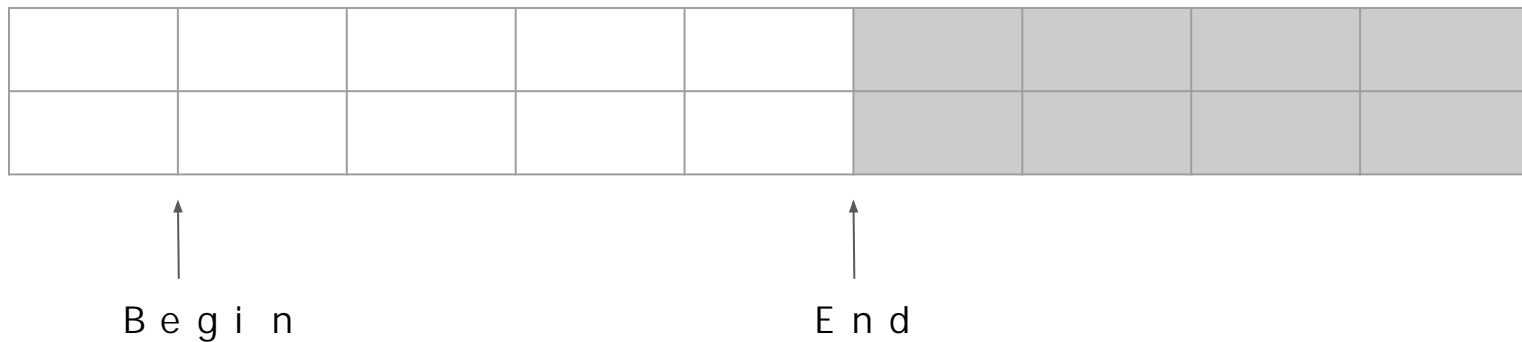
Binary Search

Example: Find the largest number less than 7



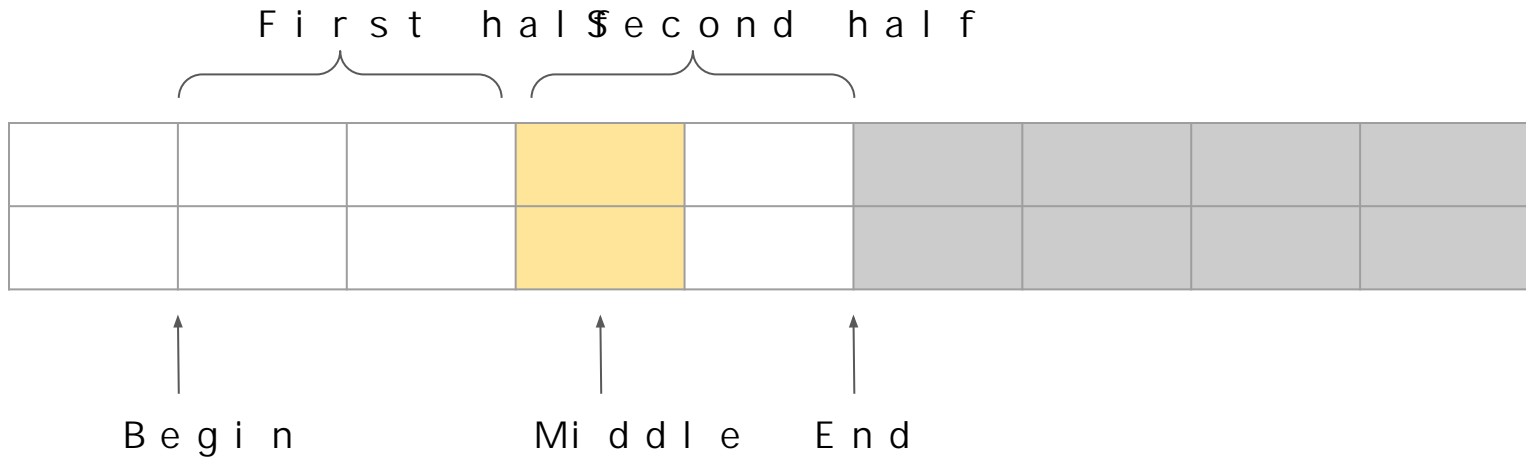
Binary Search

Example: Find the largest number less than



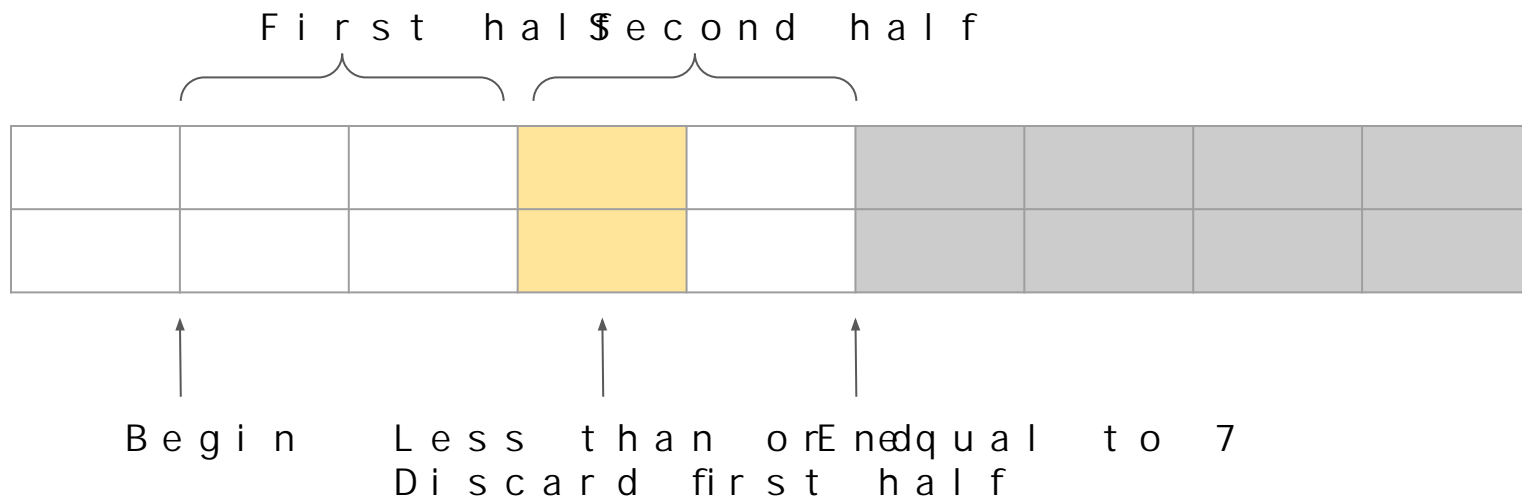
Binary Search

Example: Find the largest number less than



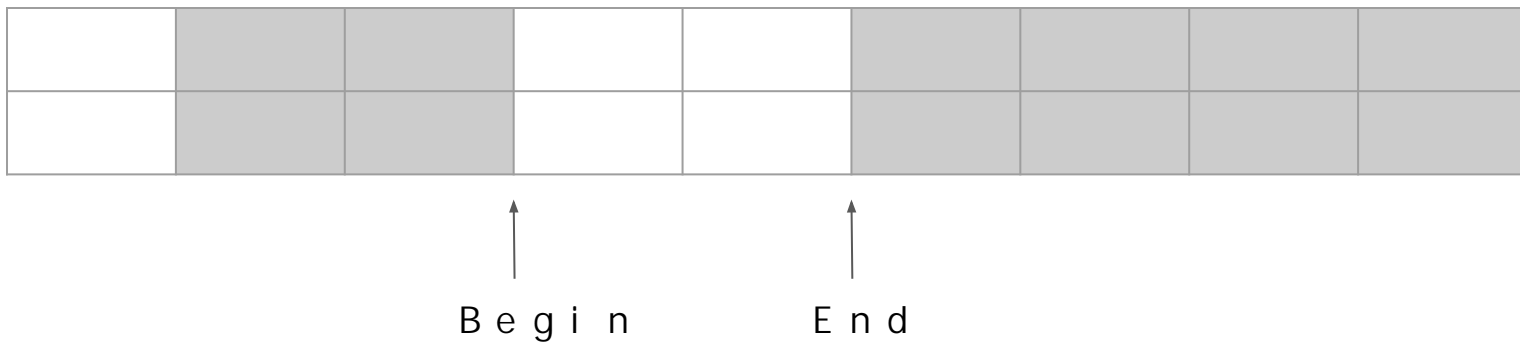
Binary Search

Example: Find the largest number less than or equal to 7



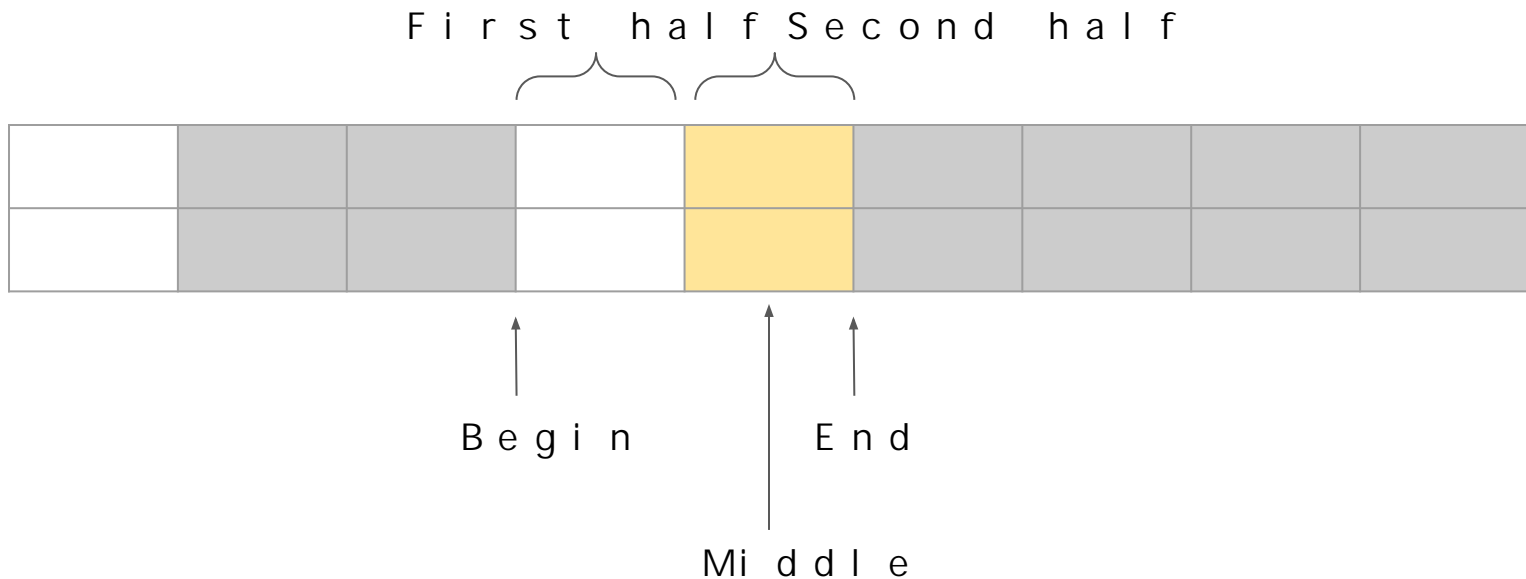
Binary Search

Example: Find the largest number less than



Binary Search

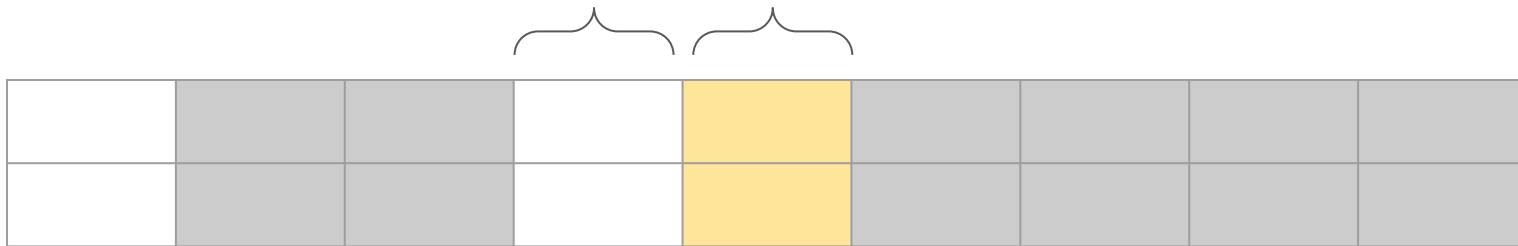
Example: Find the largest number less than



Binary Search

Example: Find the largest number less than or equal to 7

First half Second half



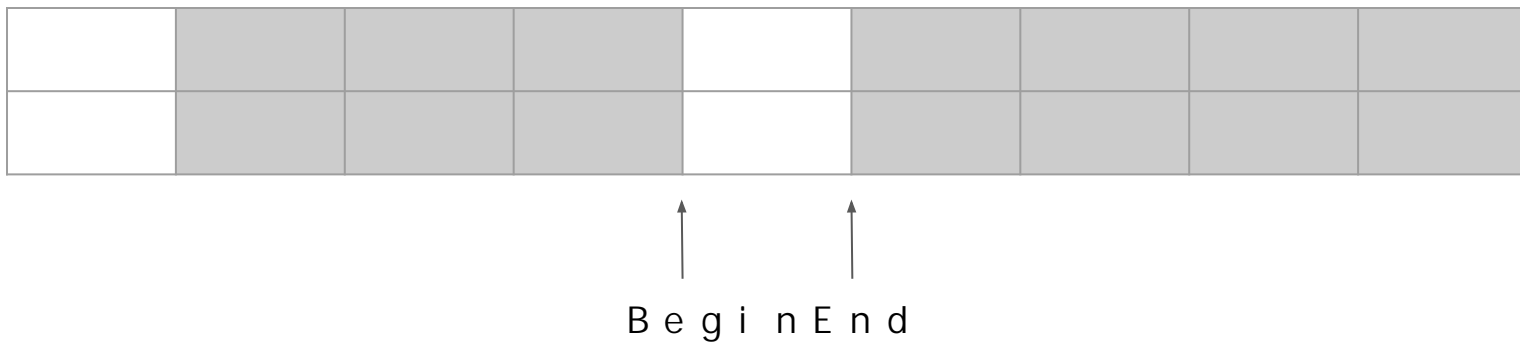
Begin

End

Less than or equal to 7
Discard first half

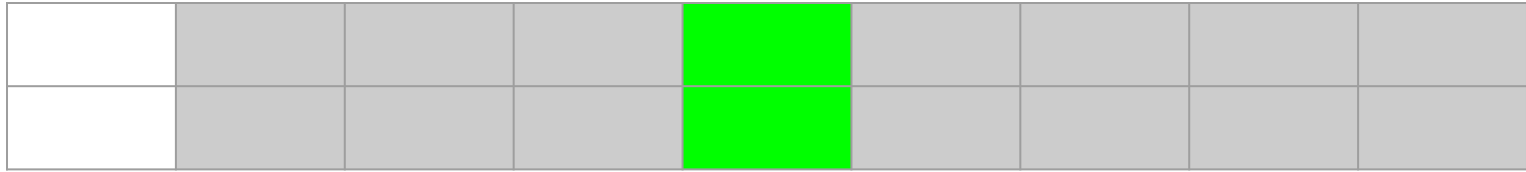
Binary Search

Example: Find the largest number less than



Binary Search

Example: Find the largest number less than or equal to 7

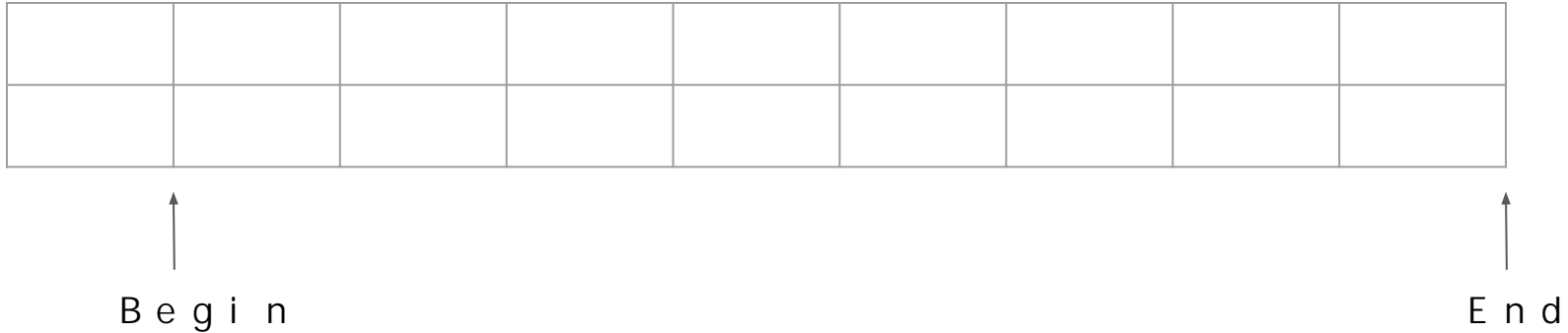


Begin n End

Less than or equal to 7
This must be the answer

Binary Search

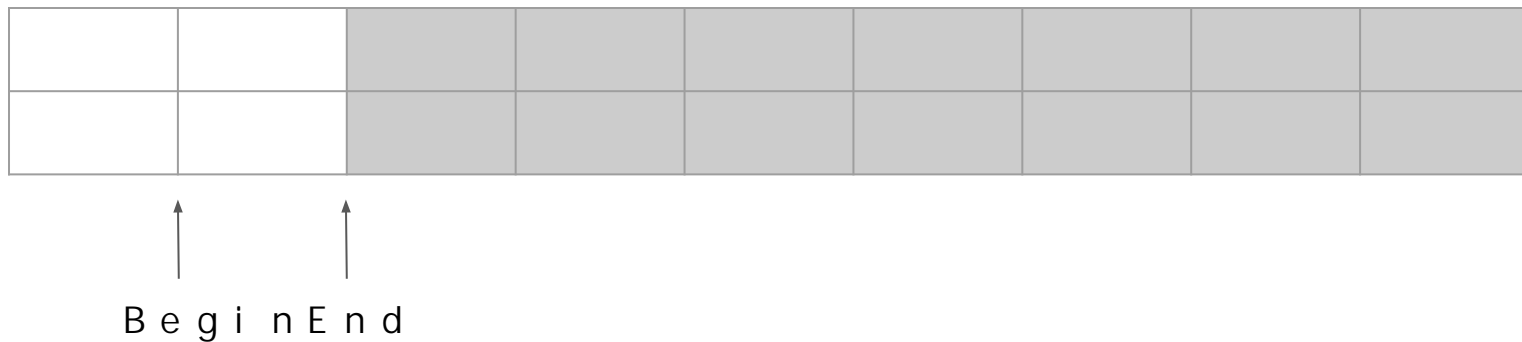
Example: Find the largest number less than



A few moments

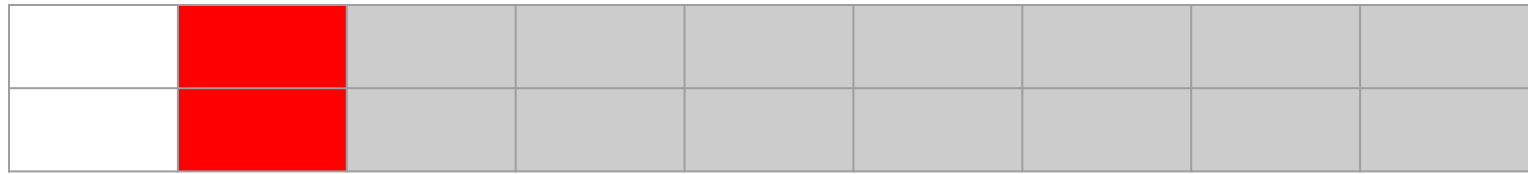
Binary Search

Example: Find the largest number less than



Binary Search

Example: Find the largest number less than



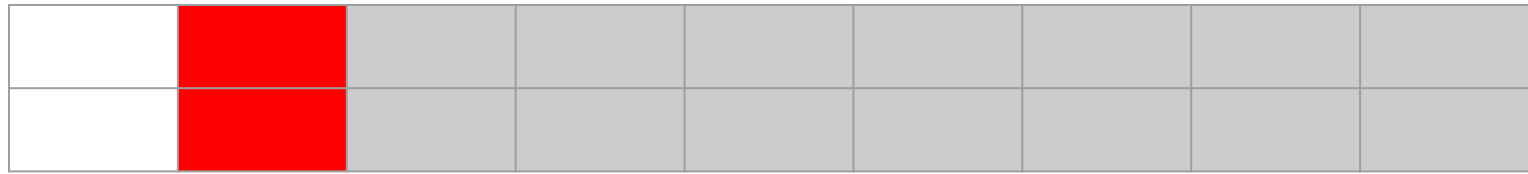
Begin n End

Greater than 1

There must be no answer

Binary Search

Example: Find the largest number less than



Begin n End

Greater than 1

There must be no answer

Many students' algorithm
just because that the
this corner case!

Binary Search

The binary search array is usually being implemented using a while loop.

Worse case: $O(\lg n)$ checking

Average: $O(\lg n)$ checking

Checking: checks whether the middle element may be the answer

```
int find(int a[], int n, int k) {
    // largest number less than or equal to k
    int bot = 0, top = n - 1; // incl
    while (bot <= top) {
        int mid = (bot + top) / 2;
        if (a[mid] <= k) {
            bot = mid + 1;
        }
        else {
            top = mid - 1;
        }
    }
    return top >= 0 ? a[top] : -1; //
}
```

Binary Search

Checklist for testing the algorithm:

sample test cases

will there be infinite loop

corner cases (no answer, answer is th

long long

...

Binary Search

Actually, there are in total 4 variations

Find the largest element less than k ~~less than k~~ The example

Find the largest element less than k

Find the smallest element greater than k

Find the smallest element greater than k



Binary Search

For C++ users, we have a good news for
The STL has already implemented two of

In the algorithm library, there are
lower_bound: find the first (smallest)
upper_bound: find the first (smallest)

Requirement: The array must be sorted

