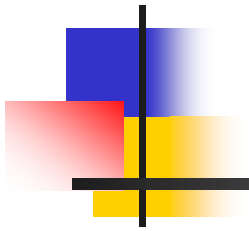
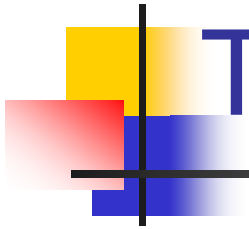


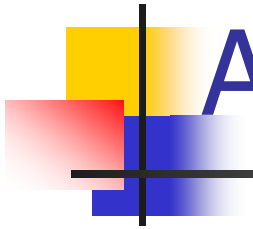
Binary Search

Alexandre David
B2-206



The Problem

- Find an element in a **sorted** array.
  - Same principle is used to find solutions of $f(x)=0$ with f continuous and monotonic.
- Divide-and-conquer algorithm:
  - Divide: Check middle element.
  - Conquer: Recursively search **1** sub-array.
  - Combine: Trivial.
  - $T(n)=T(n/2)+\Theta(1)$.



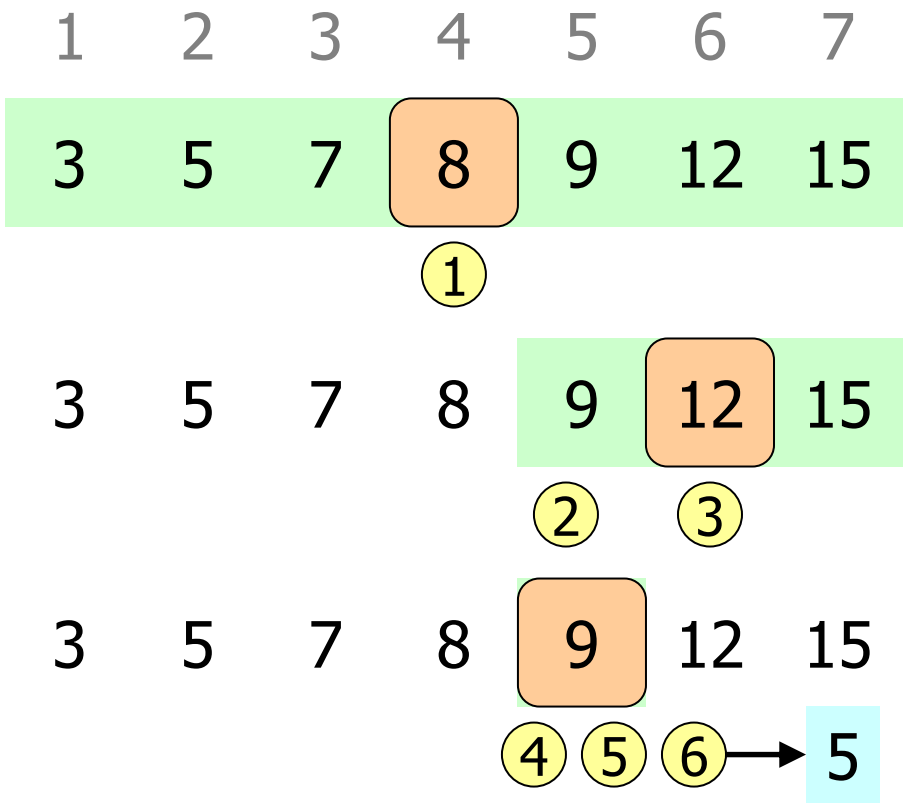
Algorithm

Iterative version of
the recursive algorithm:

```
binary_search(A,x):
min = 1
max = length(A)
repeat
    i = (min+max)/2
    if A[i] == x then
        return i
    else if A[i] < x then
        min = i+1
    else
        max = i-1
    fi
until min == max
return -1
```

Example

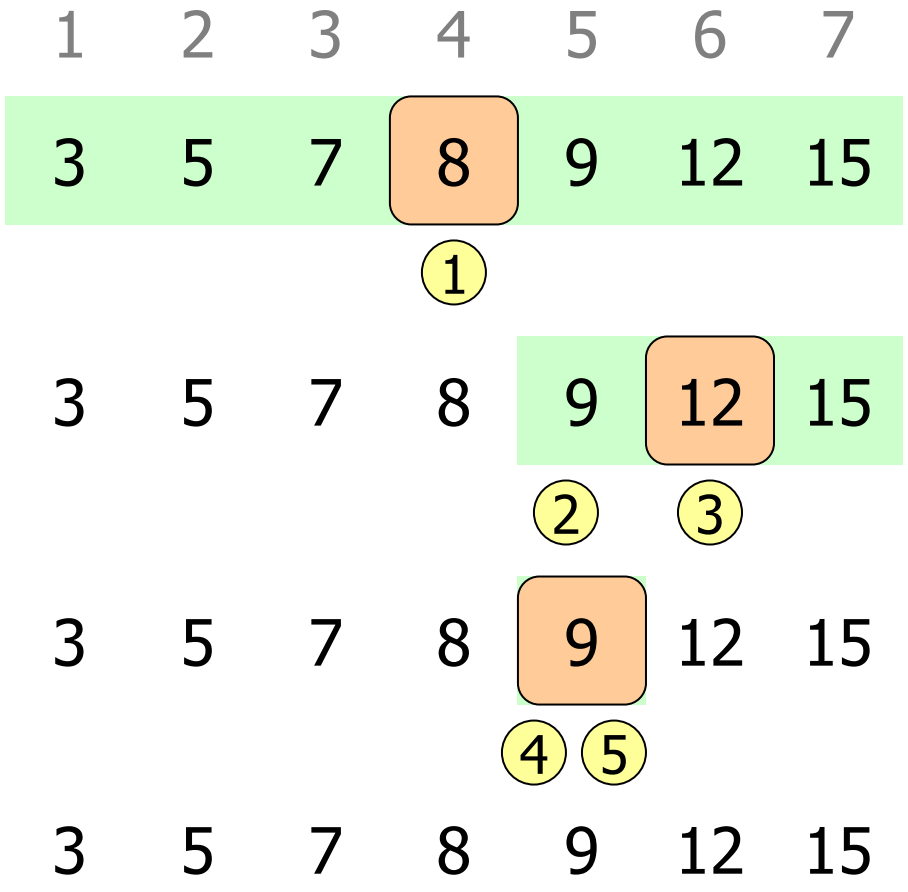
Find 9.




```
binary_search(A,x):
min = 1
max = length(A)
repeat
1 i = (min+max)/2 3 5
6 if A[i] == x then
      return i
  else if A[i] < x then
2     min = i+1
  else
4     max = i-1
  fi
until min > max
return -1
```

Example

Find 10.




```
binary_search(A,x):
min = 1
max = length(A)
repeat
① i = (min+max)/2 ③⑤
    if A[i] == x then
        return i
    else if A[i] < x then
②⑥  min = i+1
    else
④       max = i-1
    fi
until min > max
return -1          ⑦
```

Recurrence

- $T(n)=T(n/2)+\Theta(1)$.

1 sub-problem of size $n/2$.

Dividing (1) + combining (0).

- Master method, case 2 $\Rightarrow$ $T(n)=\Theta(\lg n)$.