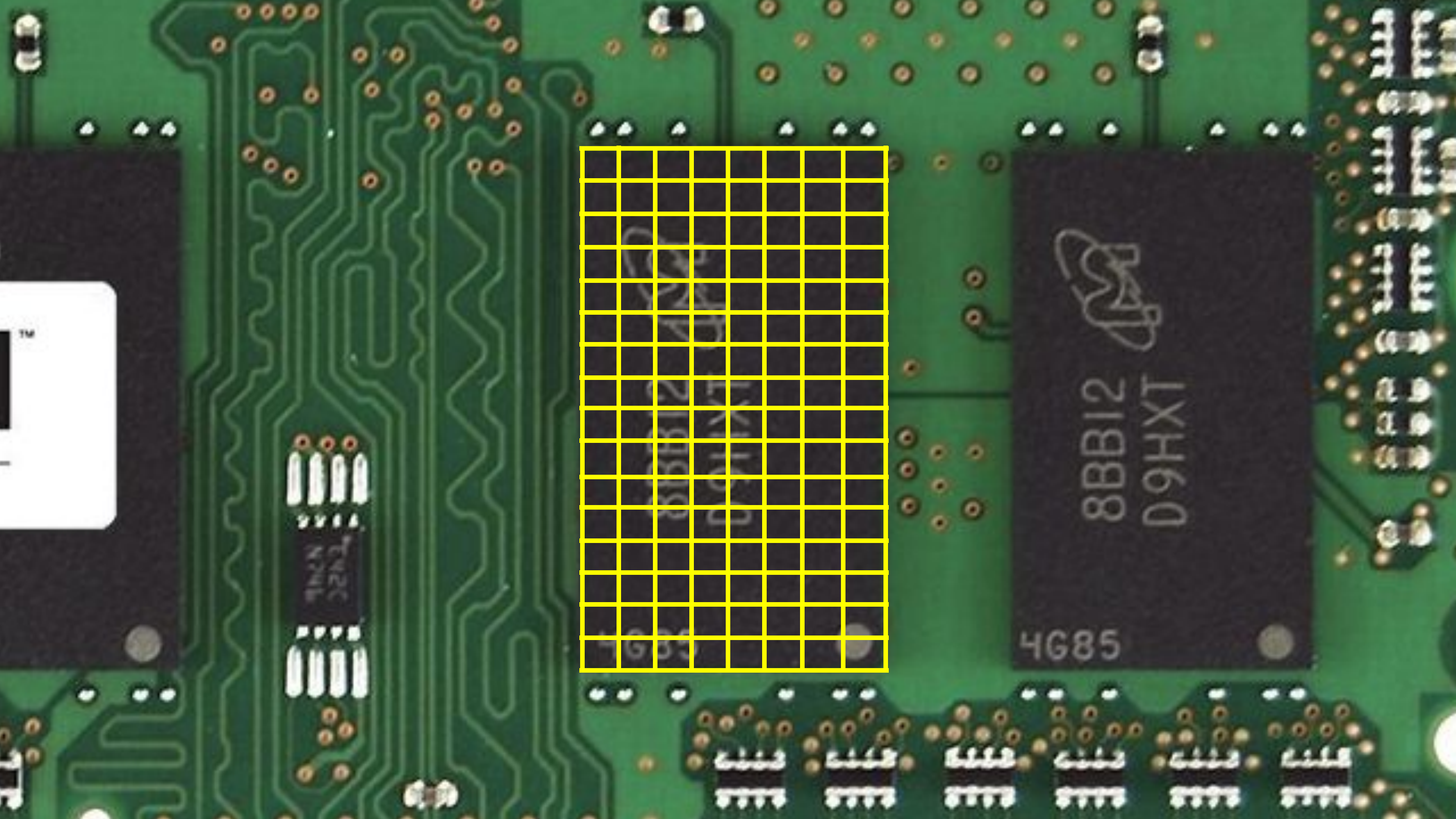
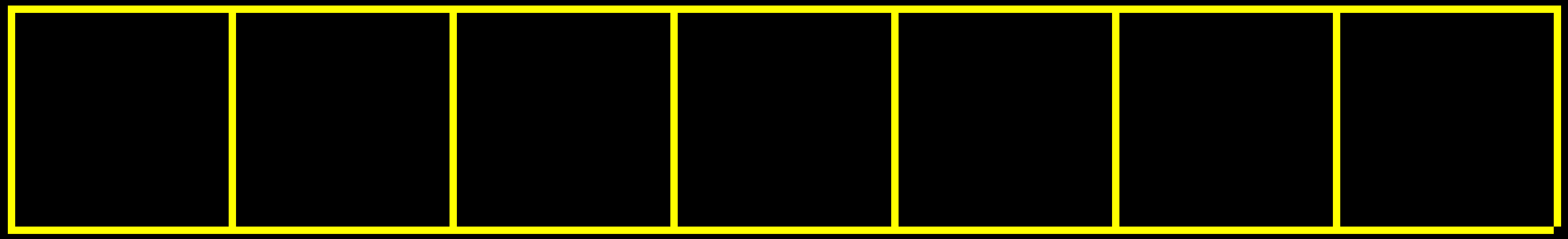
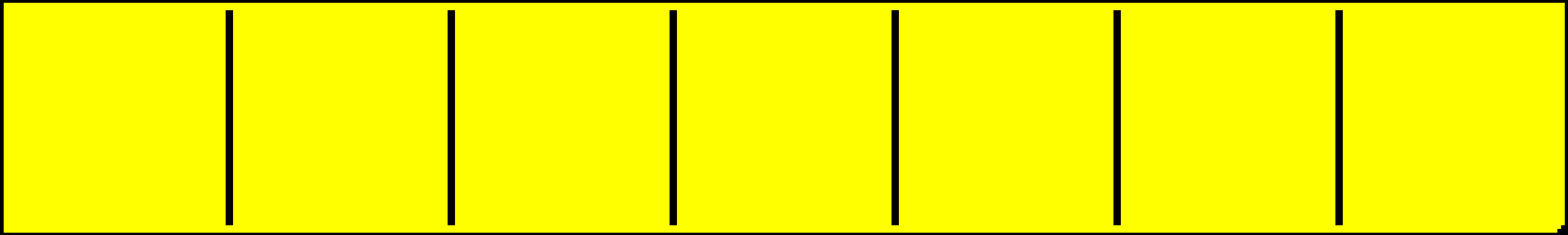
This is CS50

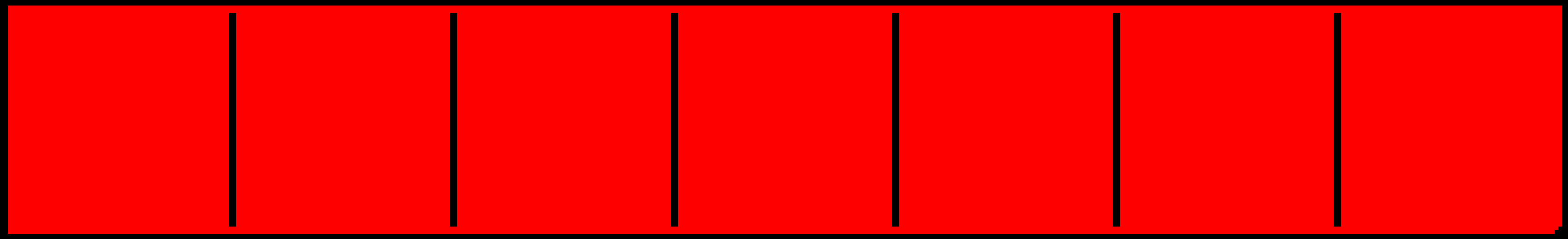

GETTING AN EDUCATION
FROM MIT
IS LIKE TRYING TO
GET A DRINK
FROM A
FIRE HOSE...

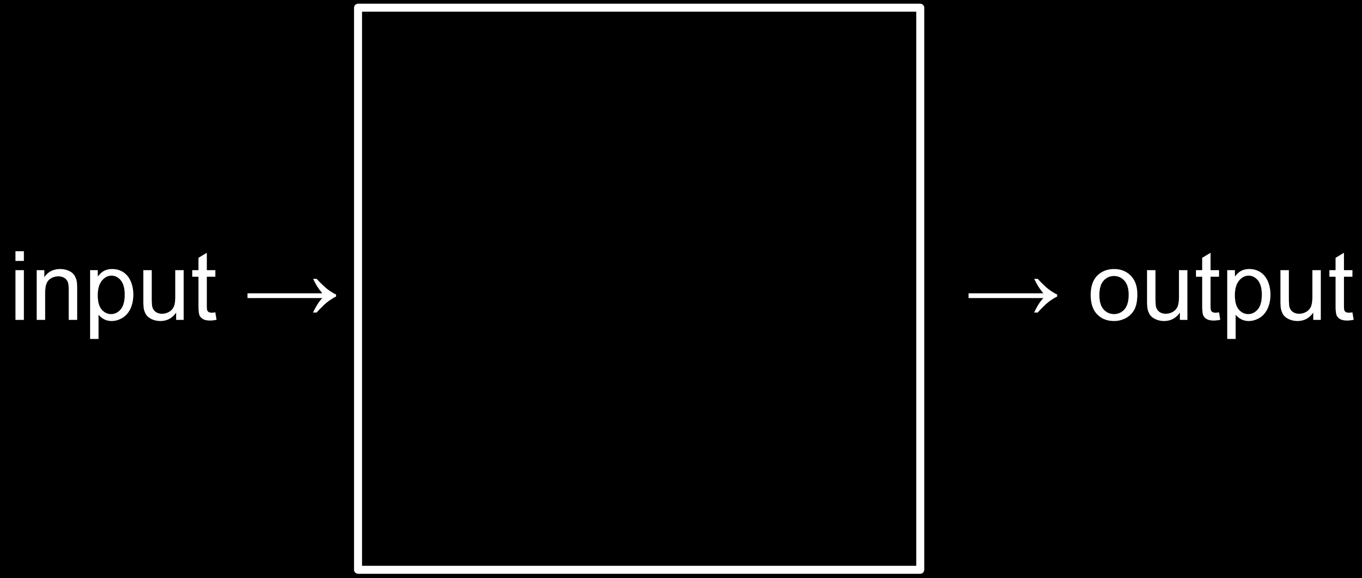

8BB12
D9HXT
4G85
8BB12
D9HXT
4G85

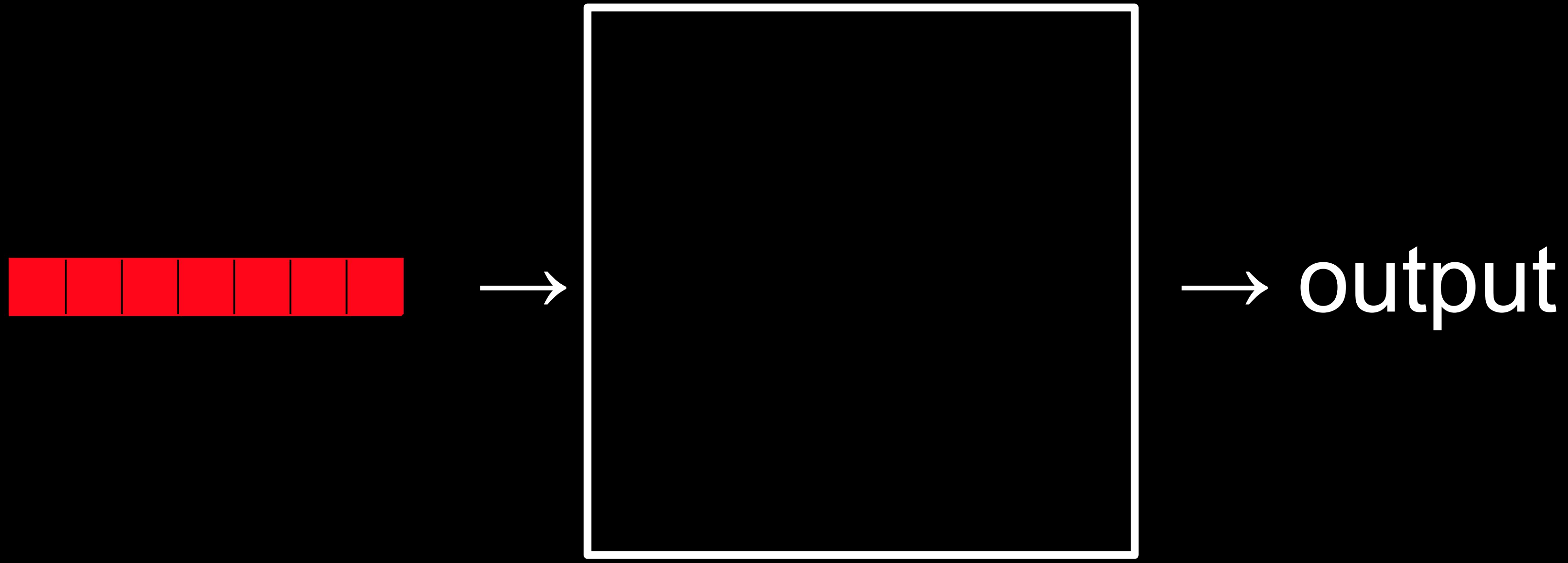

8BB12
D9HXT
4G85

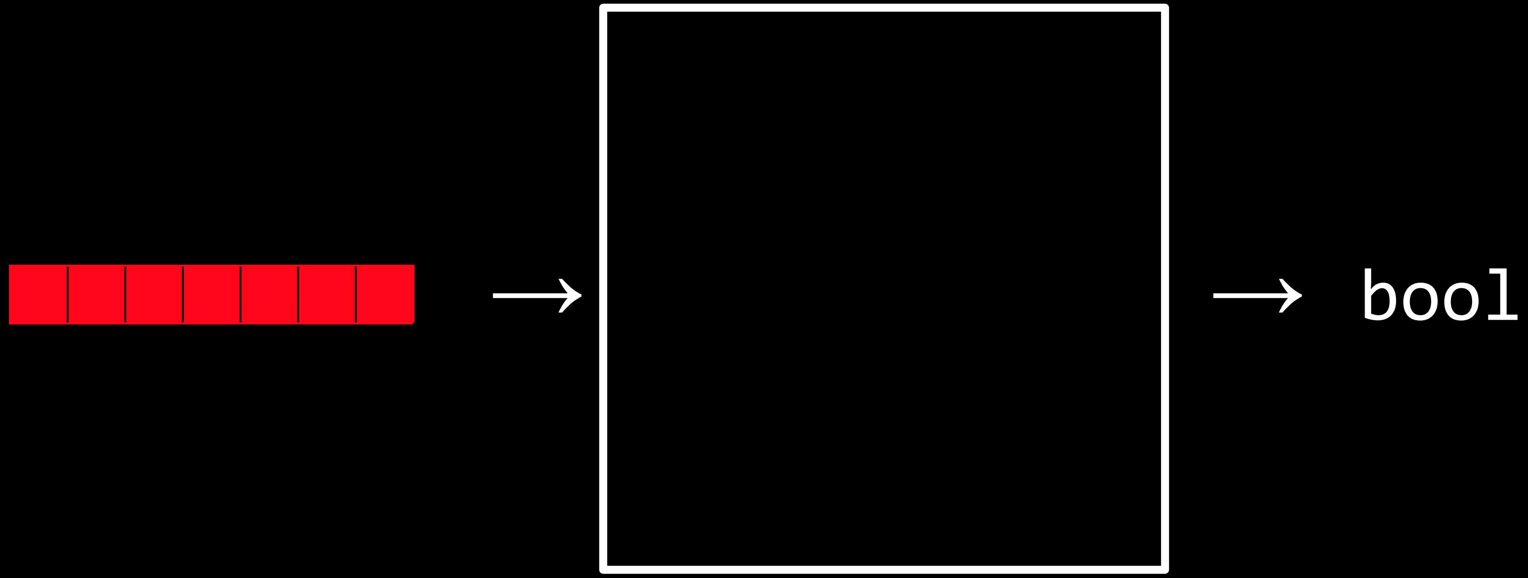







input →
→ output


→ output


bool

algorithms

linear search

```
For i from 0 to n-1
    If i'th element is 50
        Return true
Return false
```

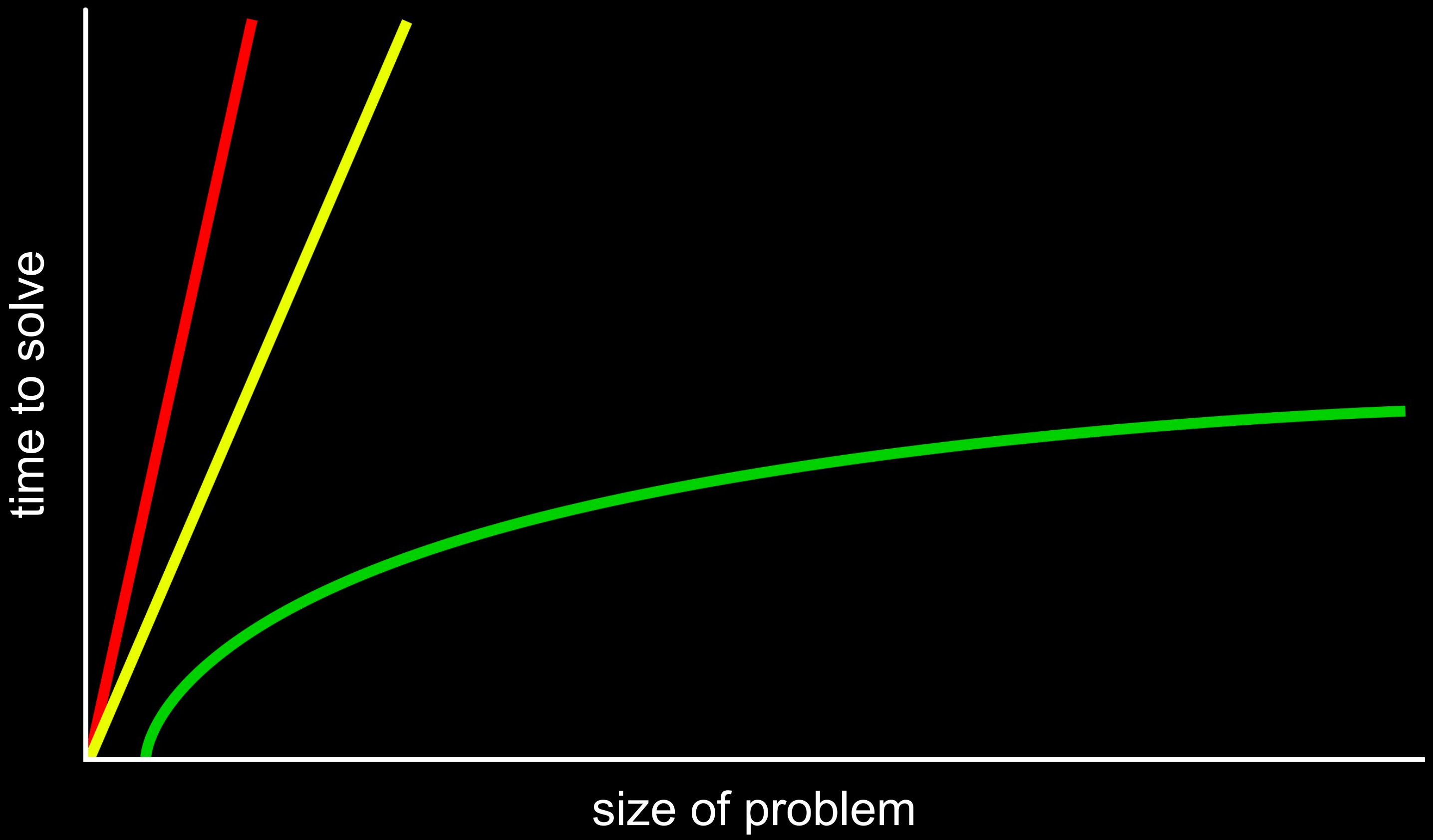
binary search

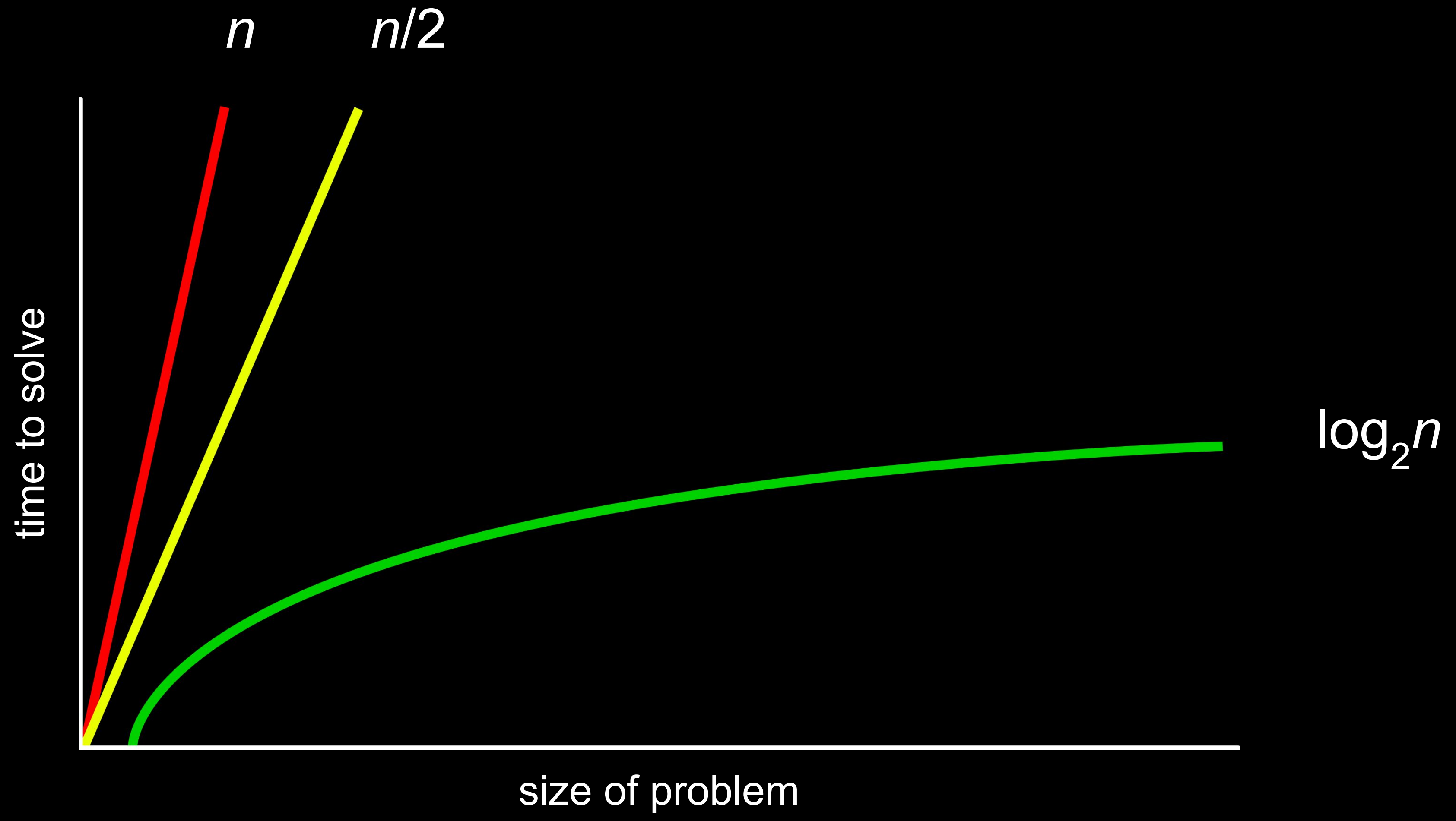
```
If middle item is 50
    Return true
Else if 50 < middle item
    Search left half
Else if 50 > middle item
    Search right half
```

```
If no items

If middle item is 50
    Return true
Else if 50 < middle item
    Search left half
Else if 50 > middle item
    Search right half
```

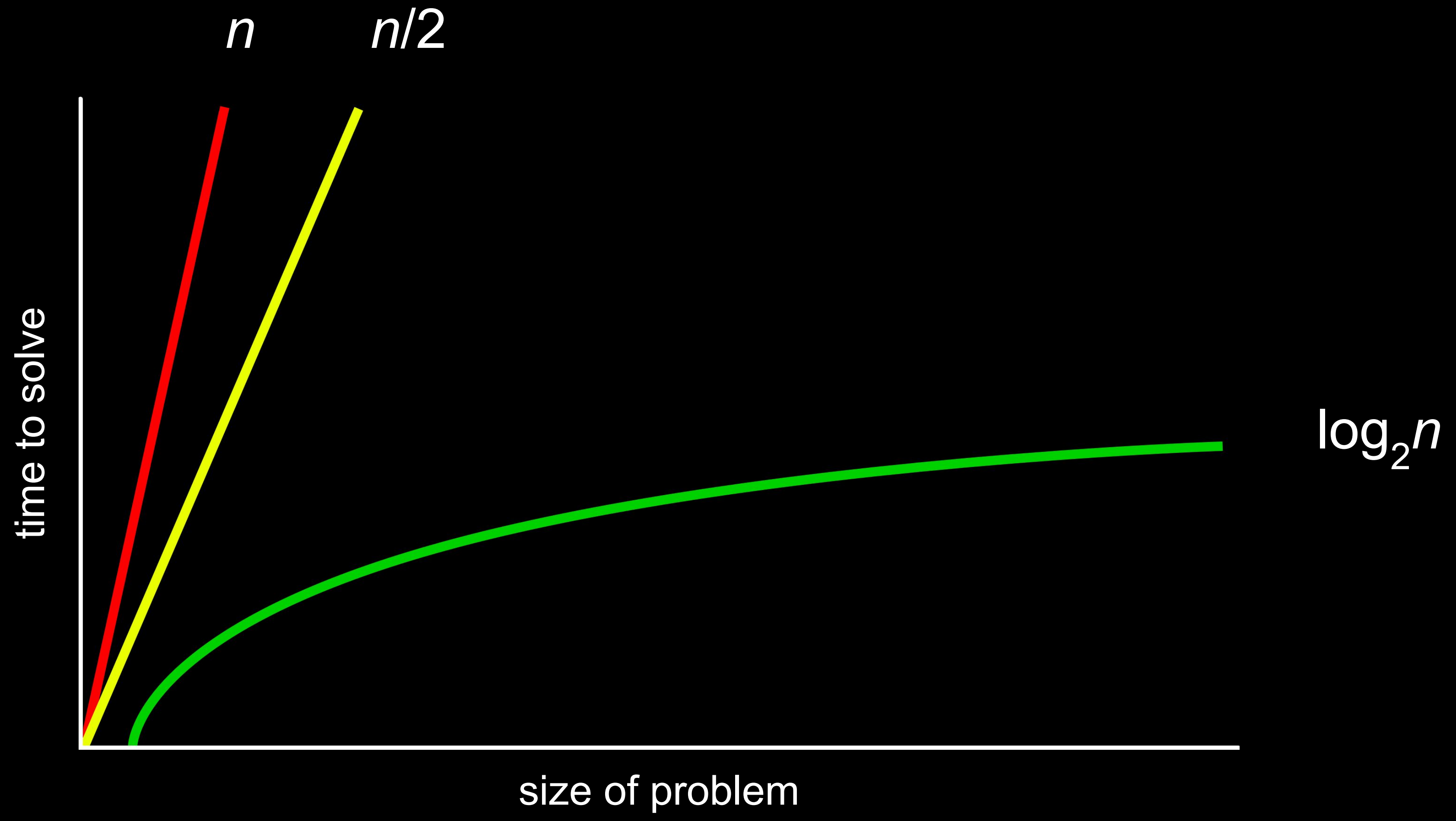
```
If no items
    Return false
If middle item is 50
    Return true
Else if 50 < middle item
    Search left half
Else if 50 > middle item
    Search right half
```


time to solve
size of problem

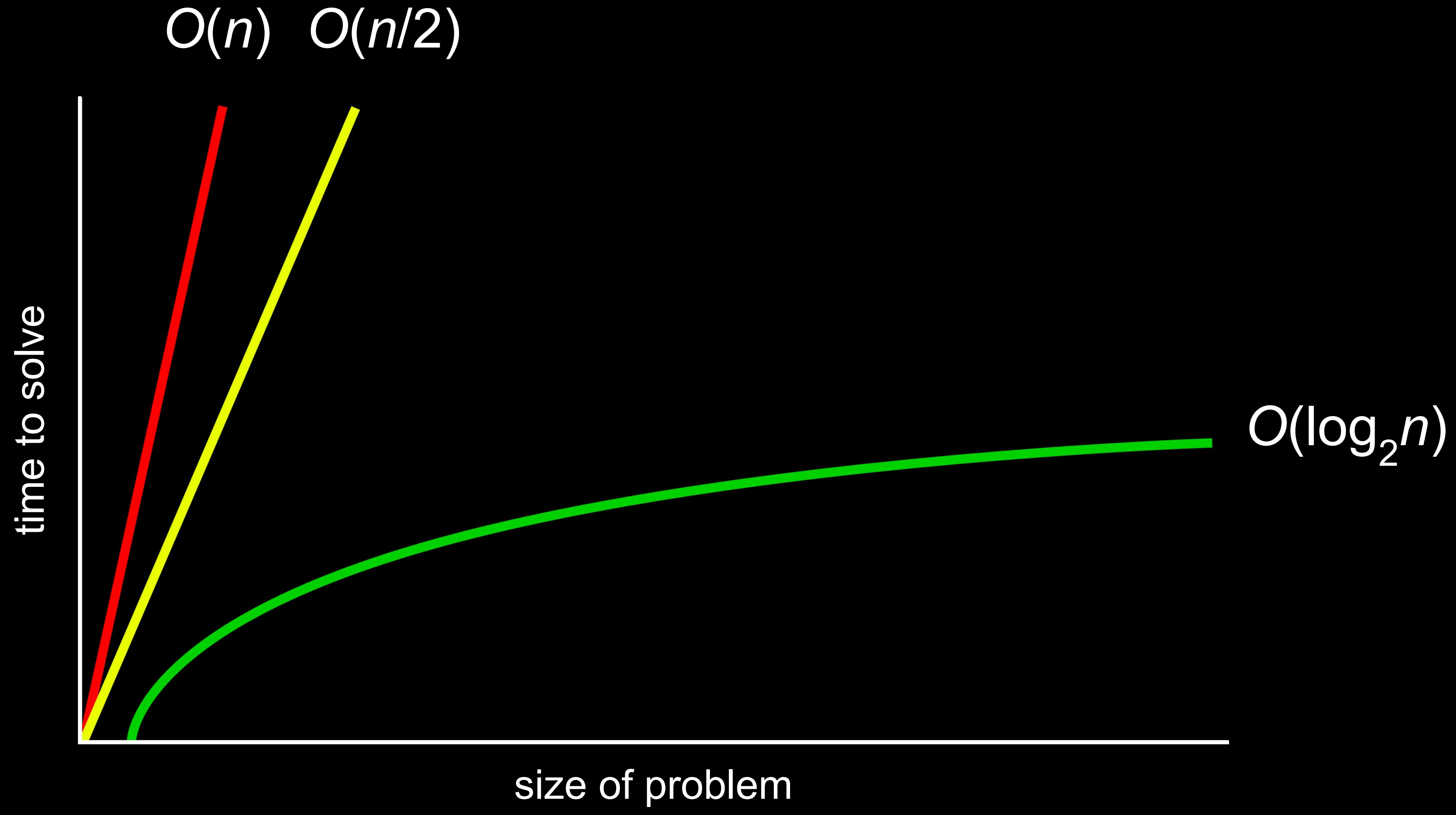


n
n/2
log2n
time to solve
size of problem

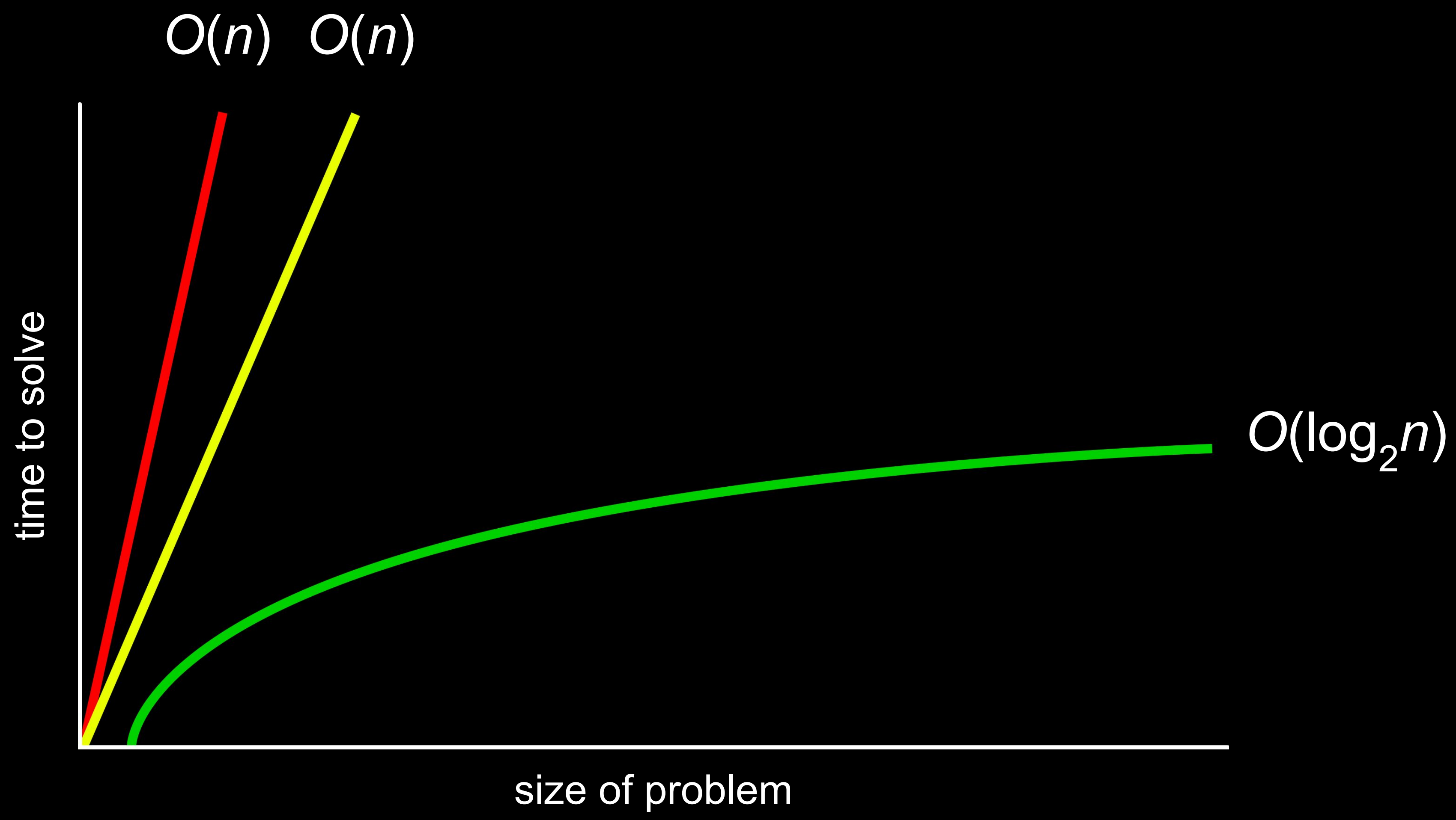
0



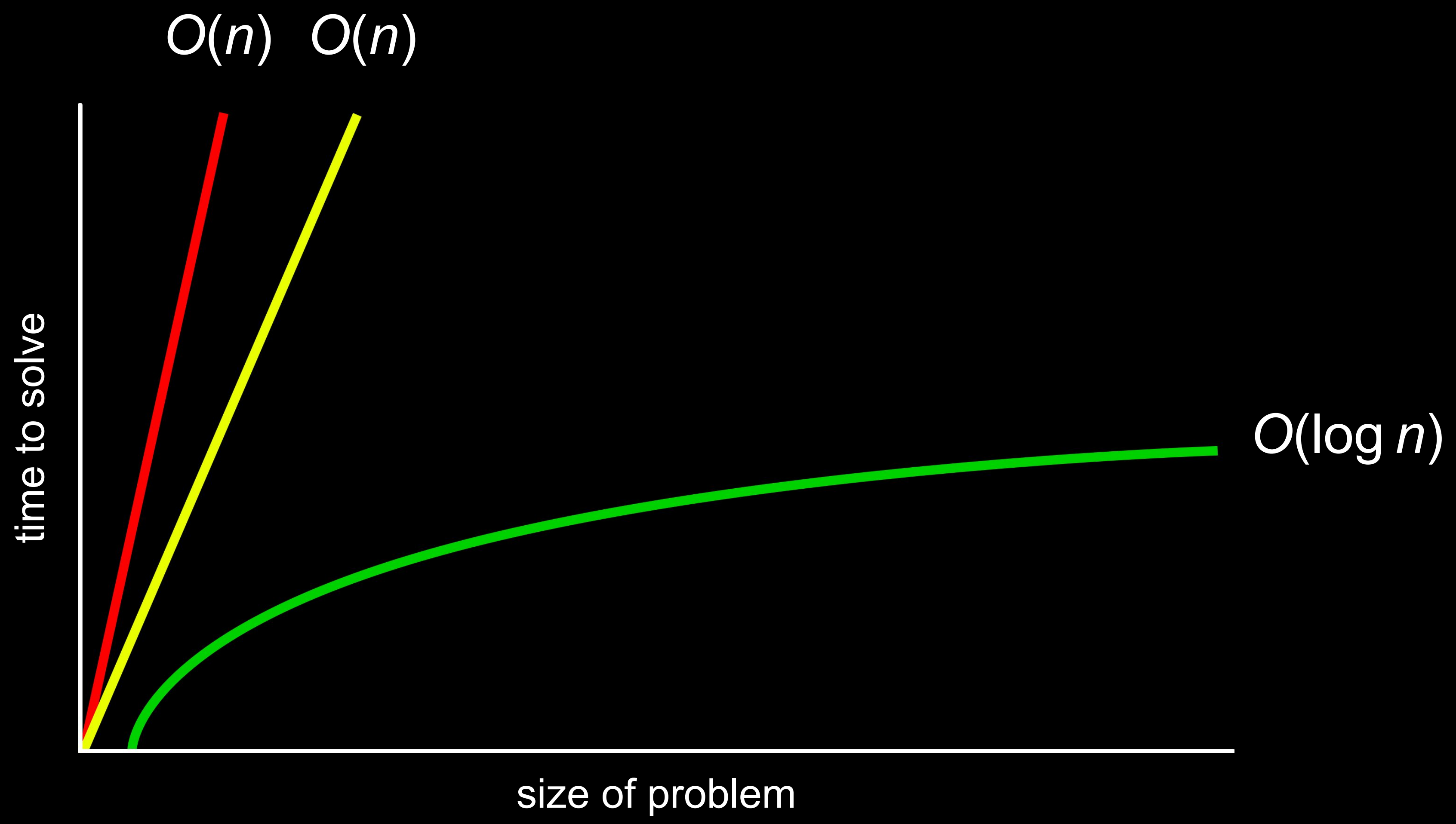
n
n/2
log2n
time to solve
size of problem

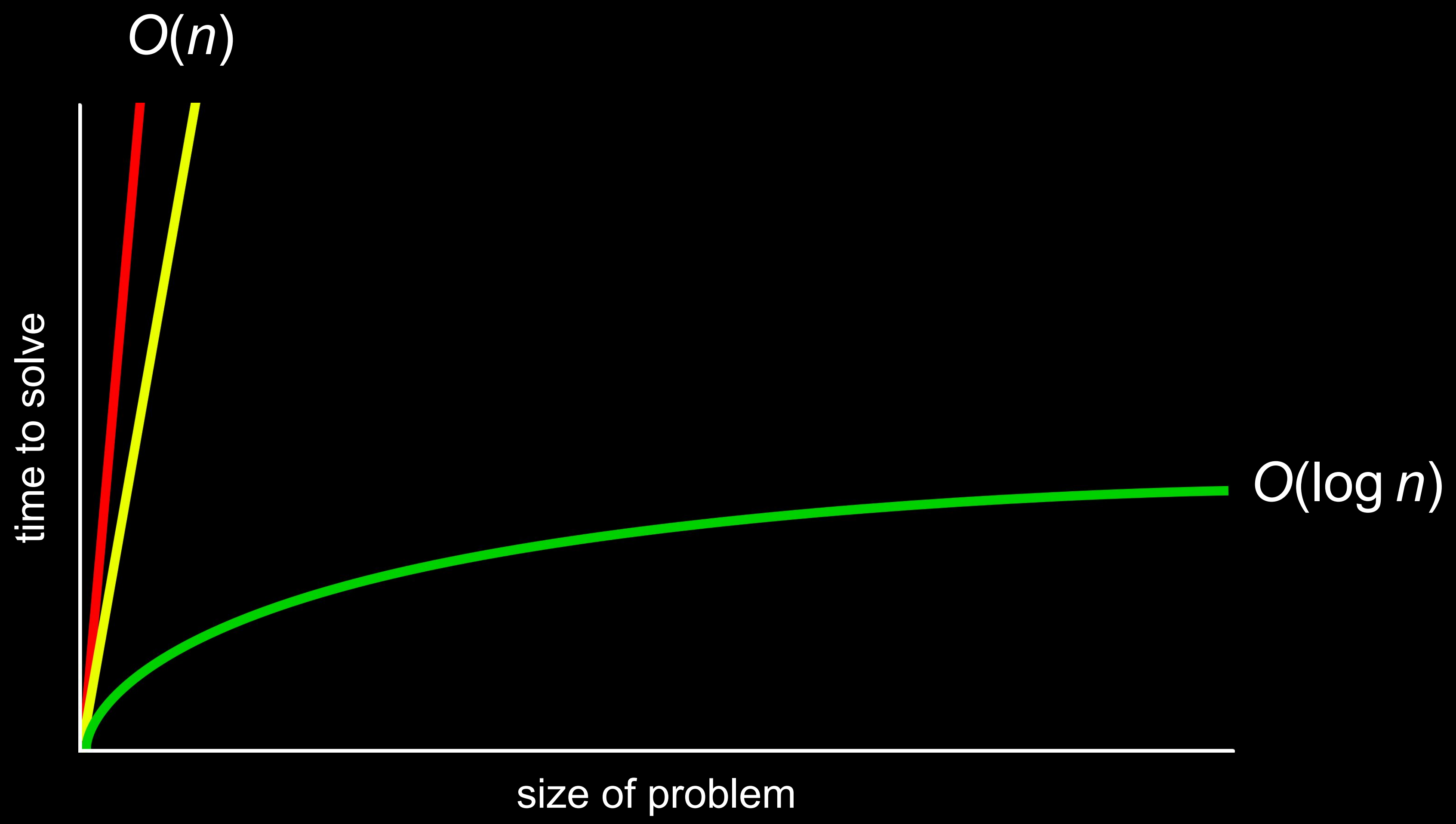


$O(n)$
$O(n/2)$
$O(\log_2 n)$
time to solve
size of problem



$O(n)$
$O(n)$
$O(\log_2 n)$
time to solve
size of problem


O(n)
O(n)
O(log n)
time to solve
size of problem



O(n)
O(log n)
time to solve
size of problem

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

$O(n^2)$

$O(n \log n)$

$O(n)$ linear search

$O(\log n)$

$O(1)$

$O(n^2)$

$O(n \log n)$

$O(n)$ linear search

$O(\log n)$ binary search

$O(1)$

Ω

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$ linear search

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$ linear search, binary search

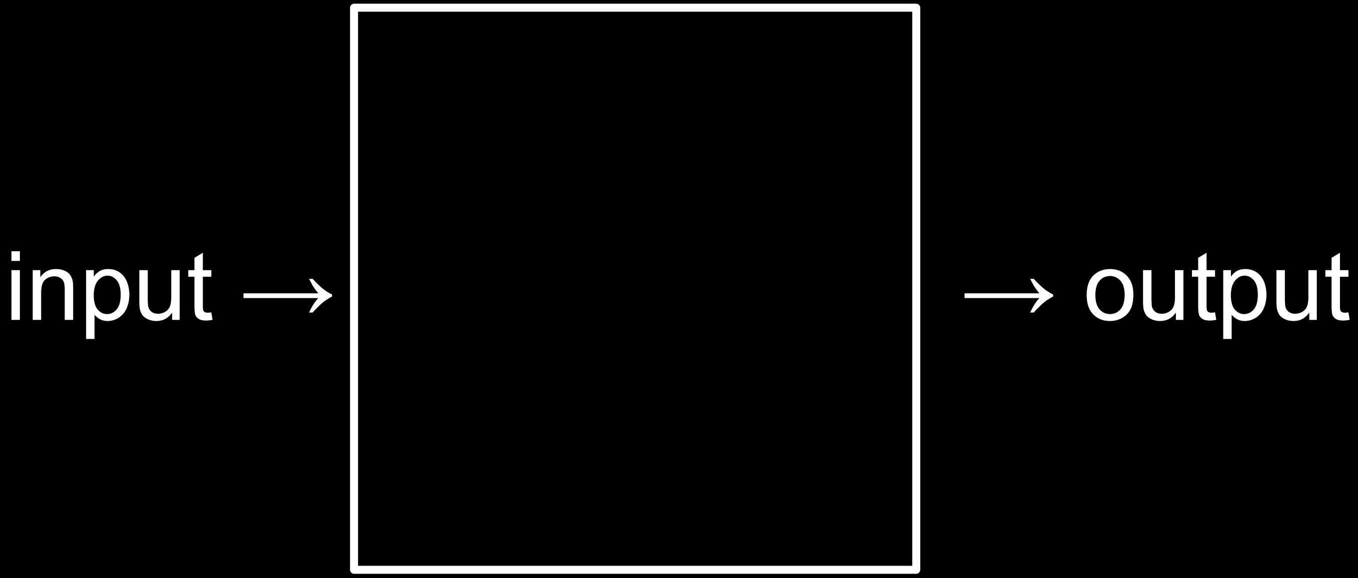
```
int numbers[]
```

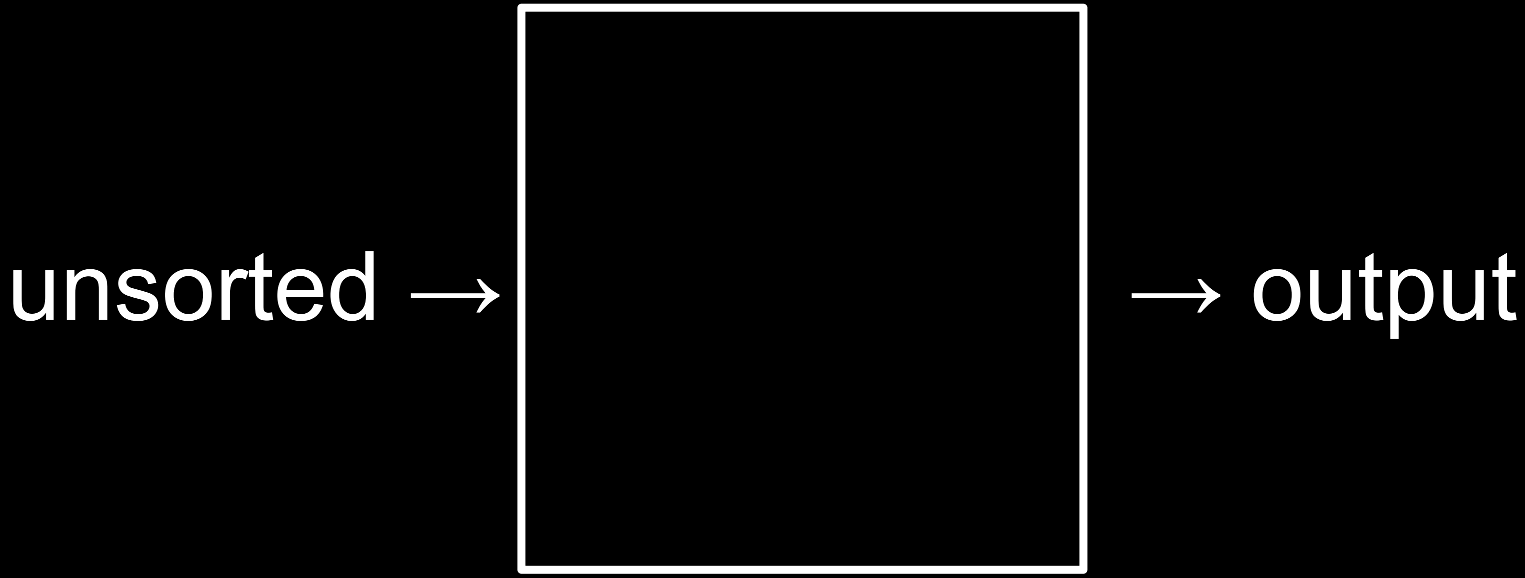
```
string names[]
```

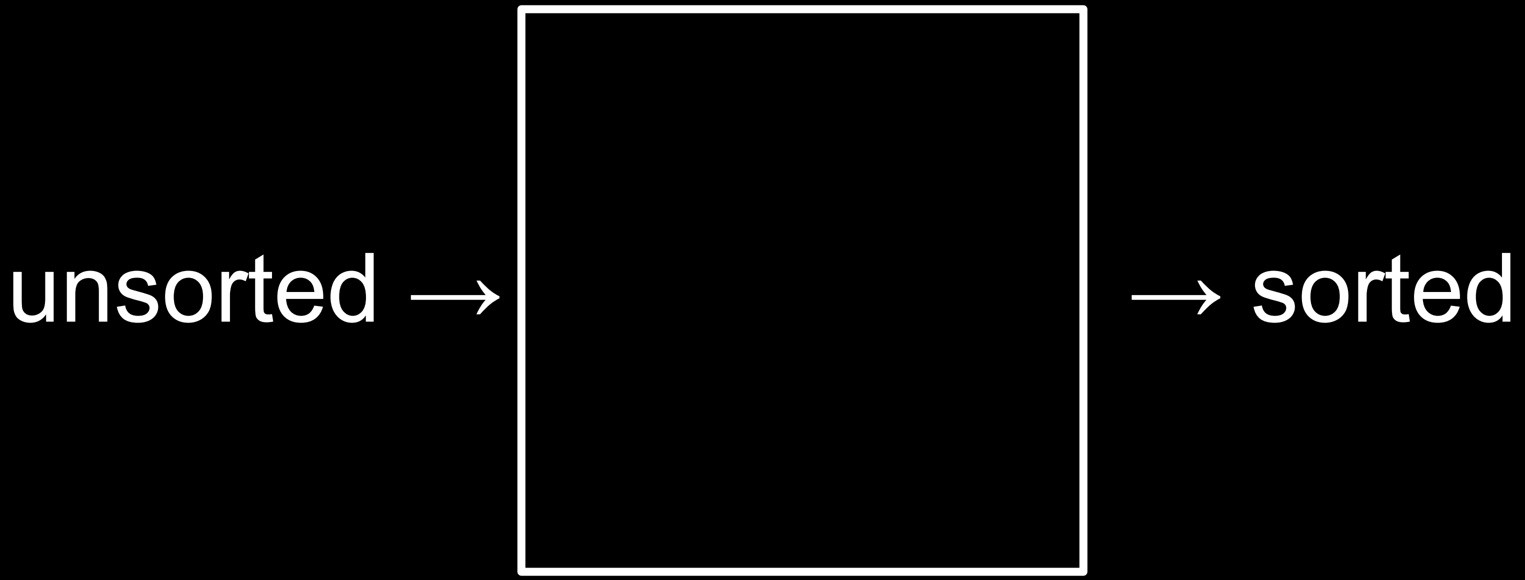
```
person people[]
```

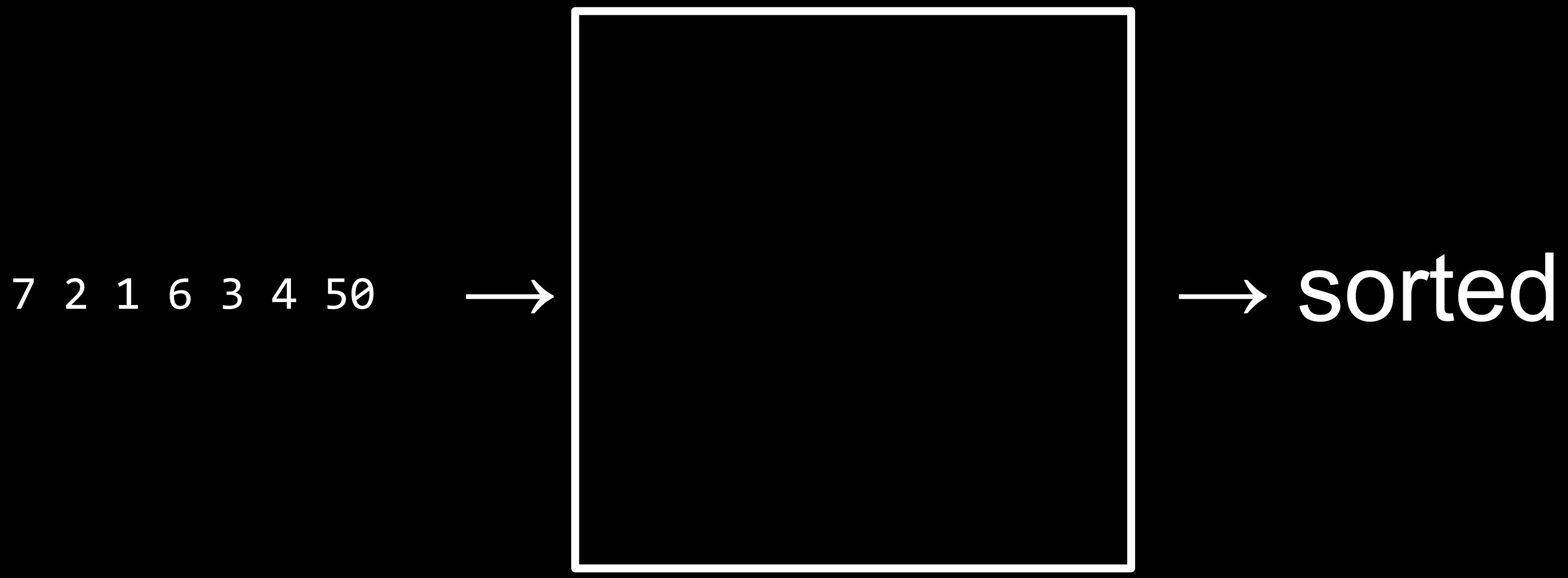
```
string name;
string number;
```

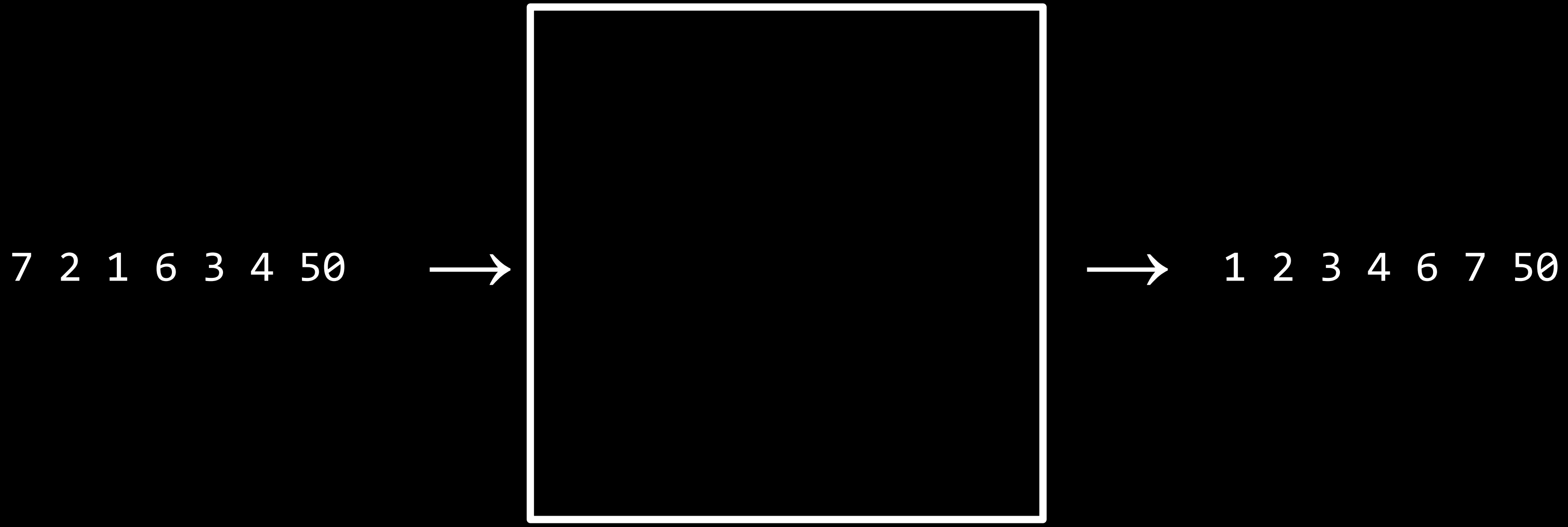
```
typedef struct
{
    string name;
    string number;
}
person;
```


input →
→ output


unsorted →
→ output


unsorted →
→ sorted


7 2 1 6 3 4 50
sorted


7 2 1 6 3 4 50
1 2 3 4 6 7 50

6 3 8 5 2 7 4 1

bubble sort

```
Repeat n-1 times
    For i from 0 to n-2
        If i'th and i+1'th elements out of order
            Swap them
```

o

```
Repeat n-1 times
    For i from 0 to n-2
        If i'th and i+1'th elements out of order
            Swap them
```

$(n - 1) \times (n - 1)$

$(n - 1) \times (n - 1)$

$n^2 - 1n - 1n + 1$

$$(n - 1) \times (n - 1)$$

$$n^2 - 1n - 1n + 1$$

$$n^2 - 2n + 1$$

$(n - 1) \times (n - 1)$

$n^2 - 1n - 1n + 1$

$n^2 - 2n + 1$

$O(n^2)$

$O(n^2)$	
$O(n \log n)$	
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

$O(n^2)$	bubble sort
$O(n \log n)$	
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

Ω

```
Repeat n-1 times
    For i from 0 to n-2
        If i'th and i+1'th elements out of order
            Swap them
```

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$ linear search, binary search

$\Omega(n^2)$	bubble sort
$\Omega(n \log n)$	
$\Omega(n)$	
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

selection sort

6 3 8 5 2 7 4 1

```
For i from 0 to n-1
    Find smallest item between i'th item and last item
    Swap smallest item with i'th item
```

O

```
For i from 0 to n-1
    Find smallest item between i'th item and last item
    Swap smallest item with i'th item
```

n

$n + (n - 1)$

$n + (n - 1) + (n - 2)$

$$n + (n - 1) + (n - 2) + \ldots + 1$$

$n + (n - 1) + (n - 2) + ... + 1$

$n(n + 1)/2$

$n + (n - 1) + (n - 2) + \ldots + 1$

$n(n + 1)/2$

$(n^2 + n)/2$

$n + (n - 1) + (n - 2) + ... + 1$

$n(n + 1)/2$

$(n^2 + \mathrm{n})/2$

$n^2/2 + n/2$

$n + (n - 1) + (n - 2) + ... + 1$

$n(n + 1)/2$

$(n^2 + \text{n})/2$

$n^2/2 + n/2$

$O(n^2)$

$O(n^2)$	bubble sort
$O(n \log n)$	
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

$O(n^2)$	bubble sort, selection sort
$O(n \log n)$	
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

Ω

```
For i from 0 to n-1
    Find smallest item between i'th item and last item
    Swap smallest item with i'th item
```

$\Omega(n^2)$	bubble sort
$\Omega(n \log n)$	
$\Omega(n)$	
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

$\Omega(n^2)$	bubble sort, selection sort
$\Omega(n \log n)$	
$\Omega(n)$	
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

bubble sort

```
Repeat n-1 times
    For i from 0 to n-2
        If i'th and i+1'th elements out of order
            Swap them
```

```
Repeat until no swaps
    For i from 0 to n-2
        If i'th and i+1'th elements out of order
            Swap them
```

$\Omega(n^2)$	bubble sort, selection sort
$\Omega(n \log n)$	
$\Omega(n)$	
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

$\Omega(n^2)$	selection sort
$\Omega(n \log n)$	
$\Omega(n)$	bubble sort
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

elections

recursion

```
1   Pick up phone book
2   Open to middle of phone book
3   Look at page
4   If Smith is on page
5       Call Mike
6   Else if Smith is earlier in book
7       Open to middle of left half of book
8       Go back to line 3
9   Else if Smith is later in book
10      Open to middle of right half of book
11      Go back to line 3
12  Else
13      Quit
```

```
1    Pick up phone book
2    Open to middle of phone book
3    Look at page
4    If Smith is on page
5        Call Mike
6    Else if Smith is earlier in book
7        Open to middle of left half of book
8        Go back to line 3
9    Else if Smith is later in book
10       Open to middle of right half of book
11       Go back to line 3
12   Else
13       Quit
```

```
Pick up phone book
Open to middle of phone book
Look at page
If Smith is on page
    Call Mike
Else if Smith is earlier in book
    Open to middle of left half of book

Else if Smith is later in book
    Open to middle of right half of book

Else
    Quit
```

```
Pick up phone book
Open to middle of phone book
Look at page
If Smith is on page
    Call Mike
Else if Smith is earlier in book
    Search left half of book

Else if Smith is later in book
    Search right half of book

Else
    Quit
```

```
Pick up phone book
Open to middle of phone book
Look at page
If Smith is on page
    Call Mike
Else if Smith is earlier in book
    Search left half of book

Else if Smith is later in book
    Search right half of book

Else
    Quit
```

```
Pick up phone book
Open to middle of phone book
Look at page
If Smith is on page
    Call Mike
Else if Smith is earlier in book
    Search left half of book
Else if Smith is later in book
    Search right half of book
Else
    Quit
```

merge sort

```
If only one item
    Return
Else
    Sort left half of items
    Sort right half of items
    Merge sorted halves
```

```
If only one item
    Return
Else
    Sort left half of items
    Sort right half of items
    Merge sorted halves
```


GETTING AN EDUCATION
FROM MIT
IS LIKE TRYING TO
GET A DRINK
FROM A
FIRE HOSE...

7 4 5 2 6 3 8 1

7	4	5	2	6	3	8	1

7 4 5 2 6 3 8 1

7	4	5	2	6	3	8	1

7 4 5 2 6 3 8 1

7	4	5	2	6	3	8	1

7	5	2	6	3	8	1
4						

5 2 6 3 8 1

4 7

5 2 6 3 8 1

4 7

5 2 6 3 8 1

4 7

5 2 6 3 8 1

4 7

5 2 6 3 8 1

4 7

5 6 3 8 1

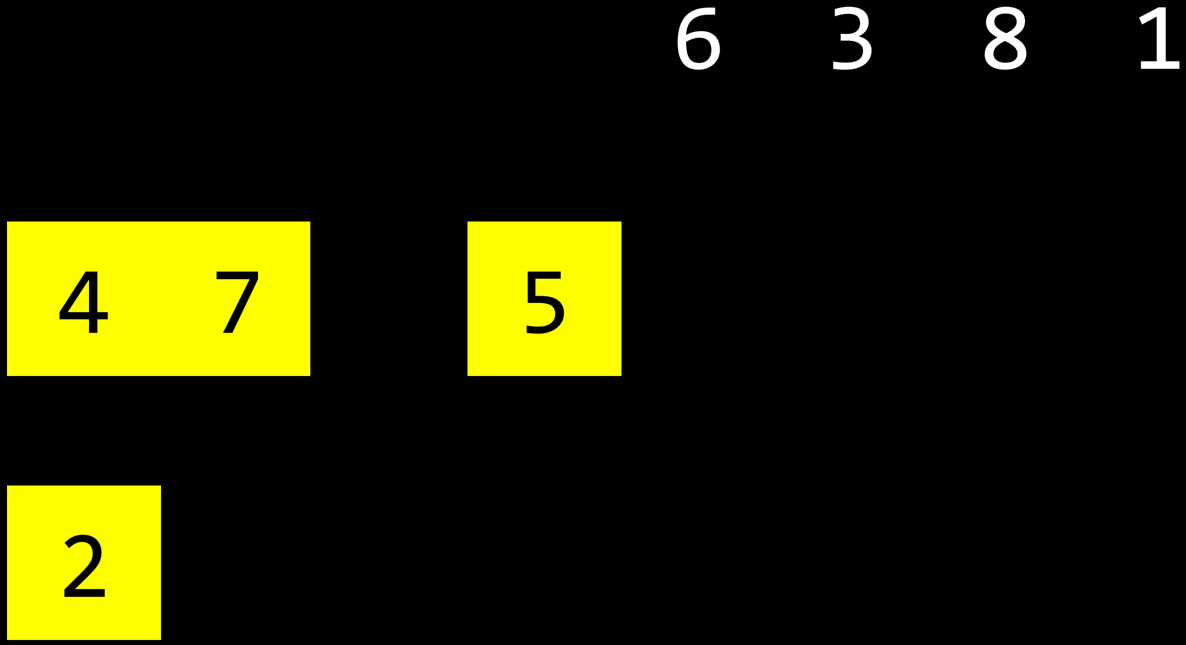
4 7 2

6 3 8 1

4 7 2 5

6 3 8 1

4	7	2	5


6 3 8 1
4 7
5
2

6 3 8 1

7 5

2 4

6 3 8 1

7

2 4 5

6 3 8 1

2 4 5 7

6 3 8 1

2 4 5 7

6 3 8 1

2 4 5 7

6 3 8 1

2 4 5 7

6 3 8 1

2 4 5 7

6 3 8 1

2 4 5 7

6 8 1

3

2 4 5 7

8 1

3 6

2 4 5 7

8 1

3 6

2 4 5 7

8 1

3 6

2 4 5 7

8 1

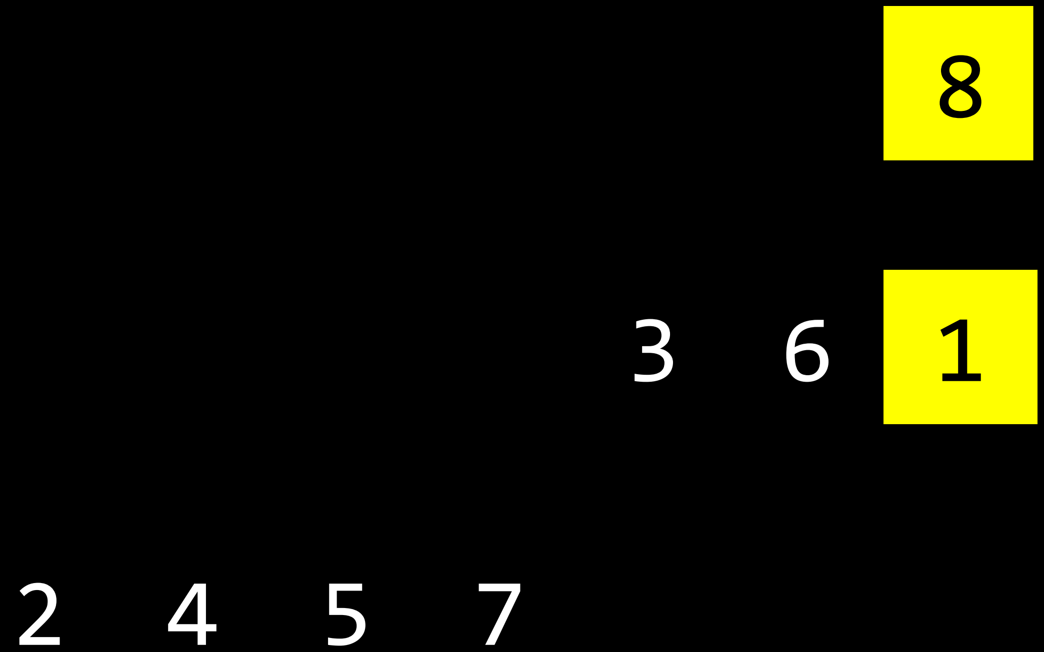
3 6

2 4 5 7

8	1

3 6

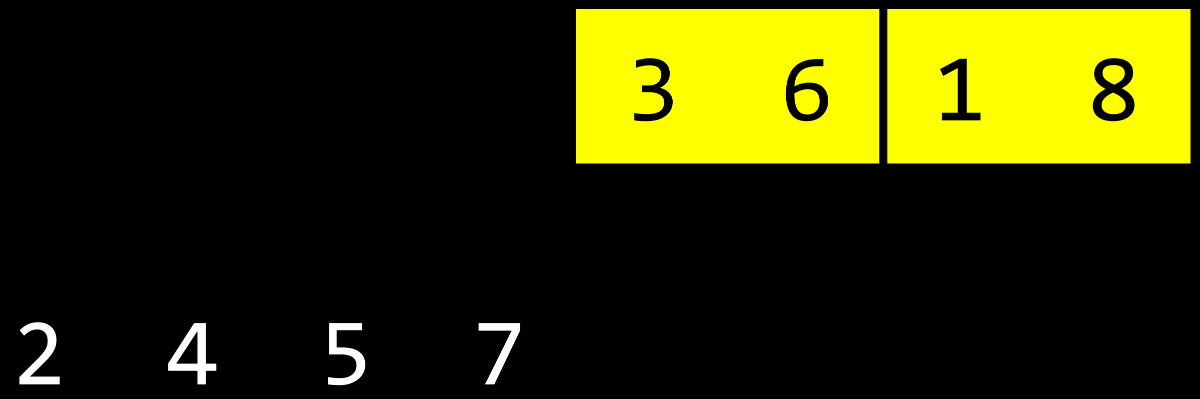
2 4 5 7



8
3
6
1
2
4
5
7

3 6 1 8

2 4 5 7


3 6 1 8
2 4 5 7

3 6 8

2 4 5 7 1

6 8

2 4 5 7 1 3

8

2 4 5 7 1 3 6

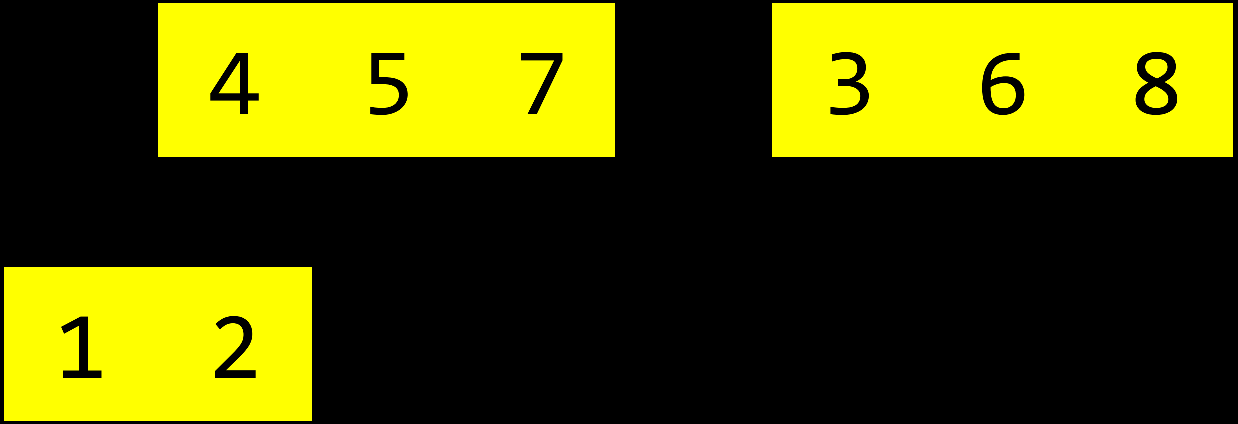
2 4 5 7 1 3 6 8

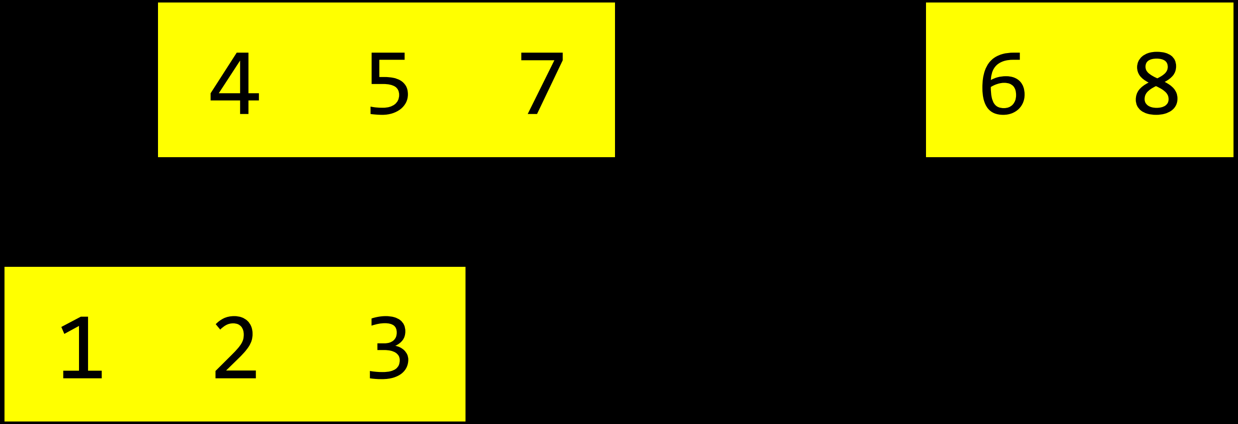
2	4	5	7	1	3	6	8

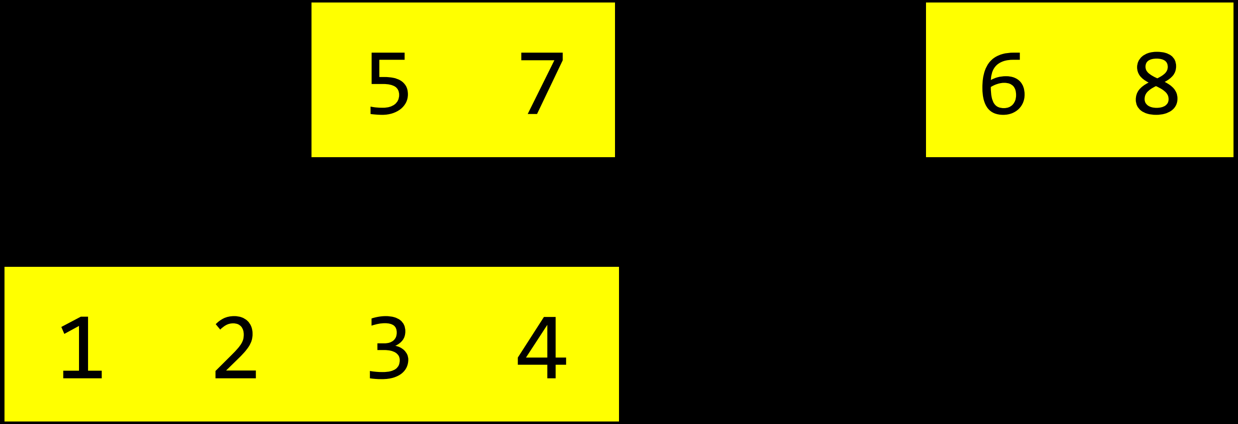
2 4 5 7

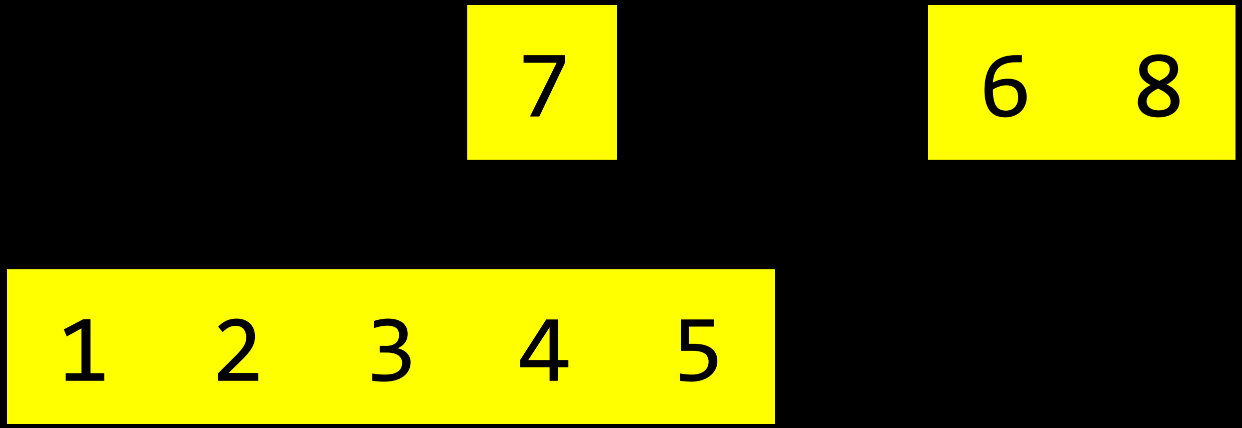
3 6 8

1


4 5 7
3 6 8
1 2


4 5 7
6 8
1 2 3


5 7
6 8
1 2 3 4


7
6 8
1 2 3 4 5

7

8

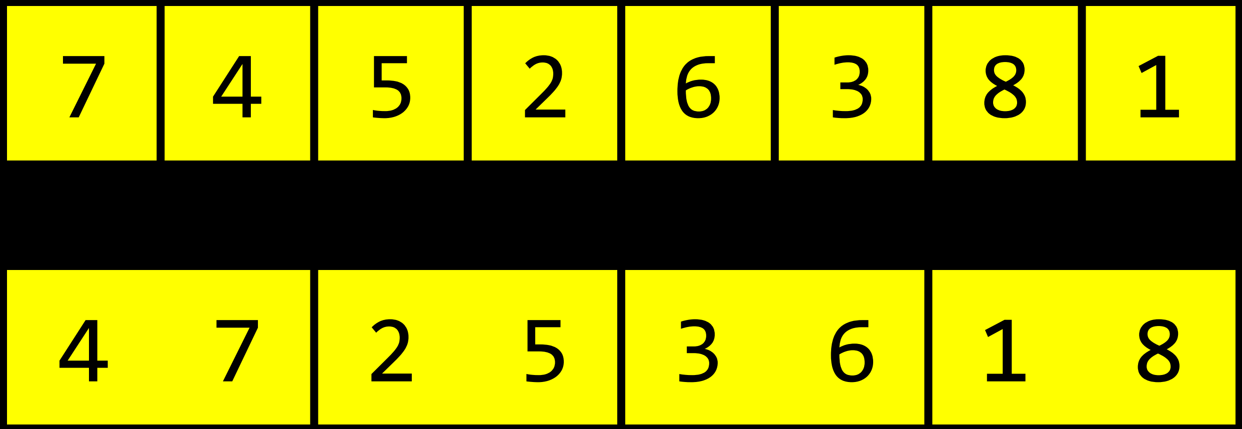
1 2 3 4 5 6

8

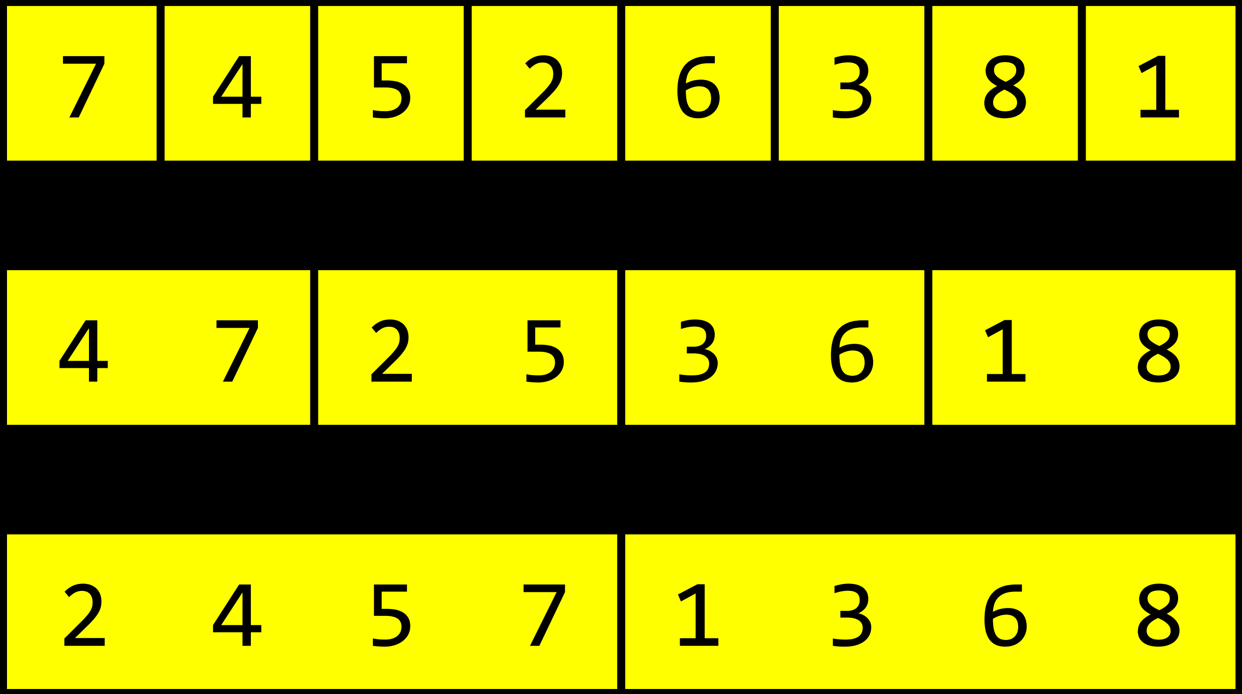
1	2	3	4	5	6	7

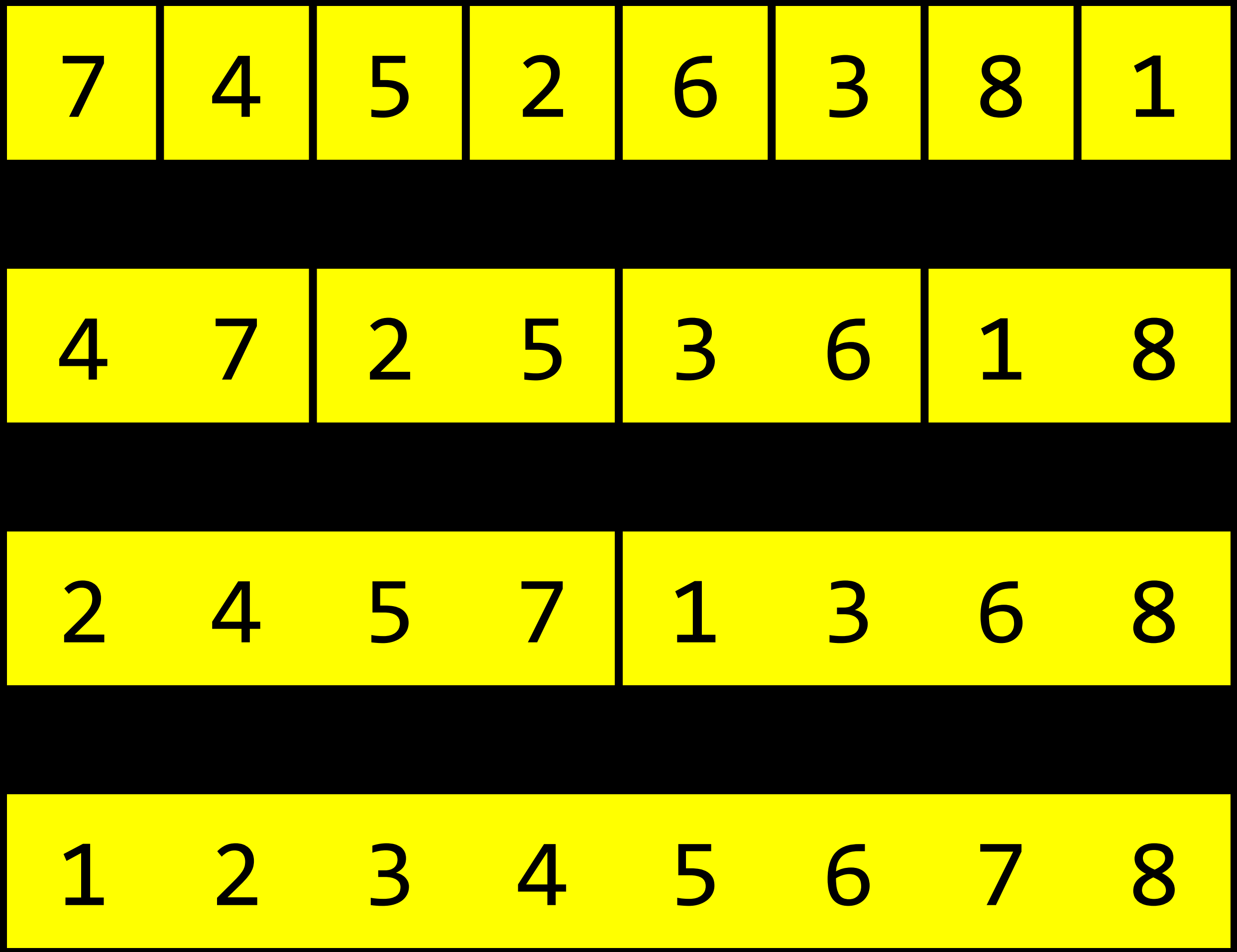
1 2 3 4 5 6 7 8

7	4	5	2	6	3	8	1



7 4 5 2 6 3 8 1
4 7 2 5 3 6 1 8


7 4 5 2 6 3 8 1
4 7 | 2 5 | 3 6 | 1 8
2 4 5 7 | 1 3 6 8


7 4 5 2 6 3 8 1
4 7 2 5 3 6 1 8
2 4 5 7 1 3 6 8
1 2 3 4 5 6 7 8

$O(n^2)$	bubble sort, selection sort
$O(n \log n)$	
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

$O(n^2)$	bubble sort, selection sort
$O(n \log n)$	merge sort
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

$\Omega(n^2)$	selection sort
$\Omega(n \log n)$	
$\Omega(n)$	bubble sort
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

$\Omega(n^2)$	selection sort
$\Omega(n \log n)$	merge sort
$\Omega(n)$	bubble sort
$\Omega(\log n)$	
$\Omega(1)$	linear search, binary search

Θ

$\Theta(n^2)$

$\Theta(n \log n)$

$\Theta(n)$

$\Theta(\log n)$

$\Theta(1)$

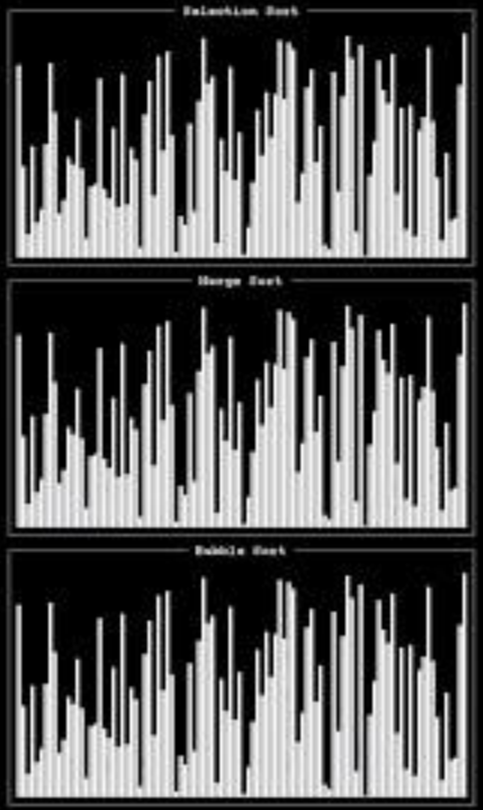
$\Theta(n^2)$ selection sort

$\Theta(n \log n)$ merge sort

$\Theta(n)$

$\Theta(\log n)$

$\Theta(1)$


Selection Sort
Merge Sort
Bubble Sort

This is CS50