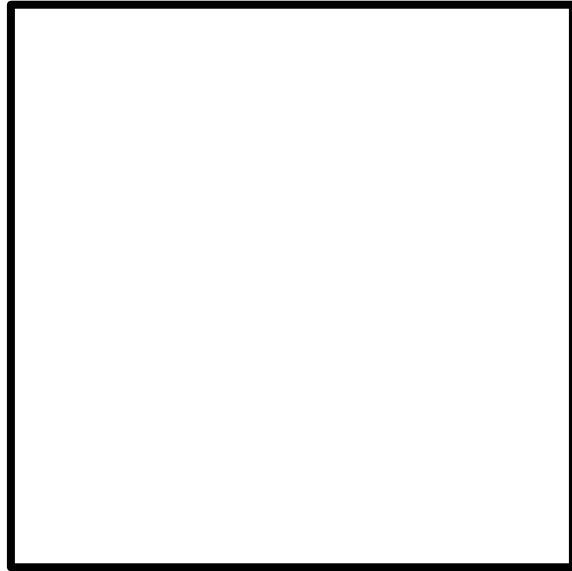


CS50 for JDs

Algorithms, Data Structures

shorts

input →



→ output



algorithms





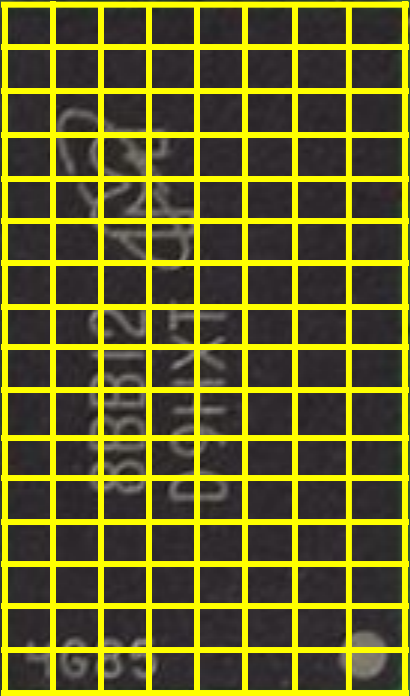
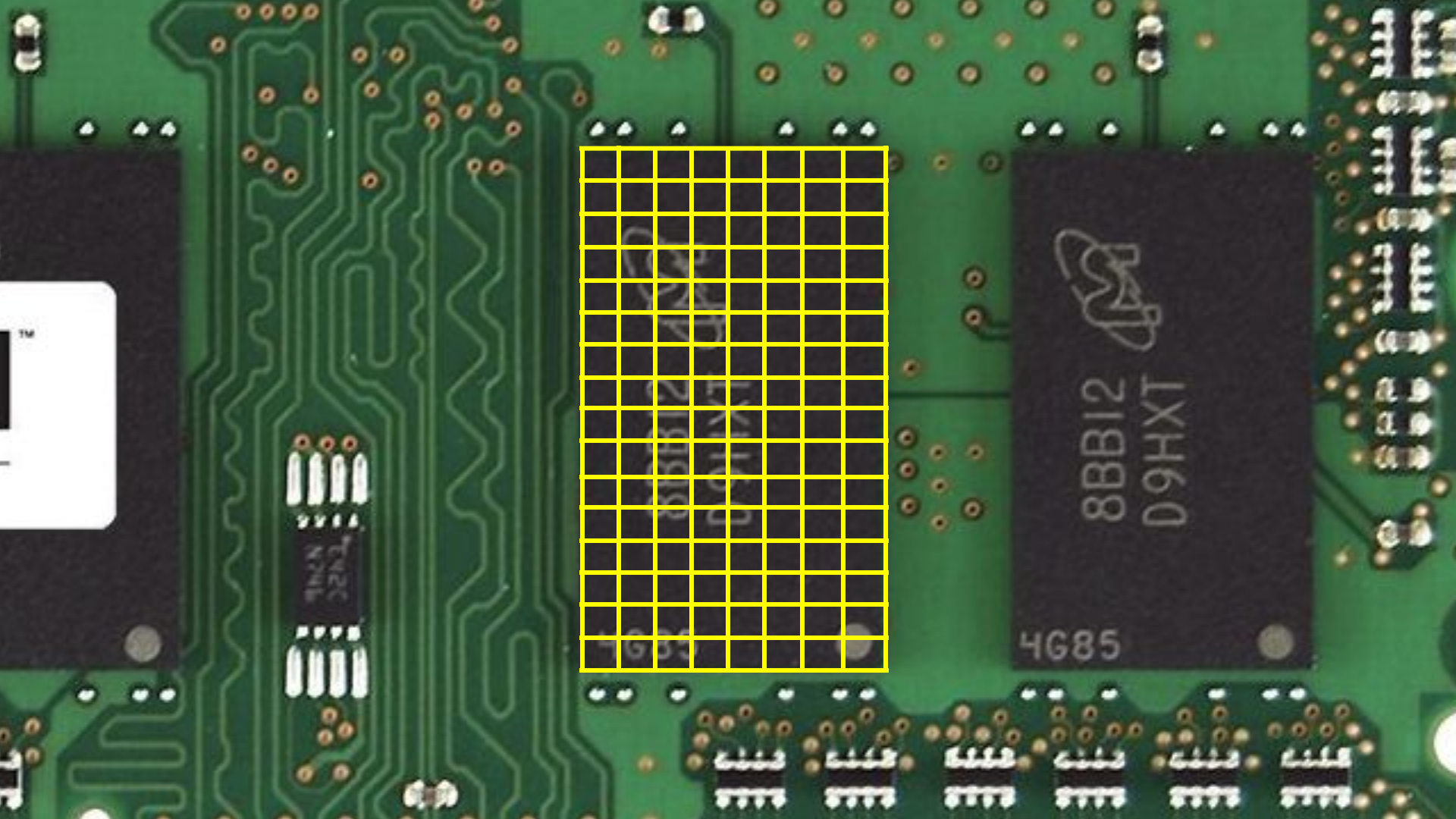
8BB12
D9HXT

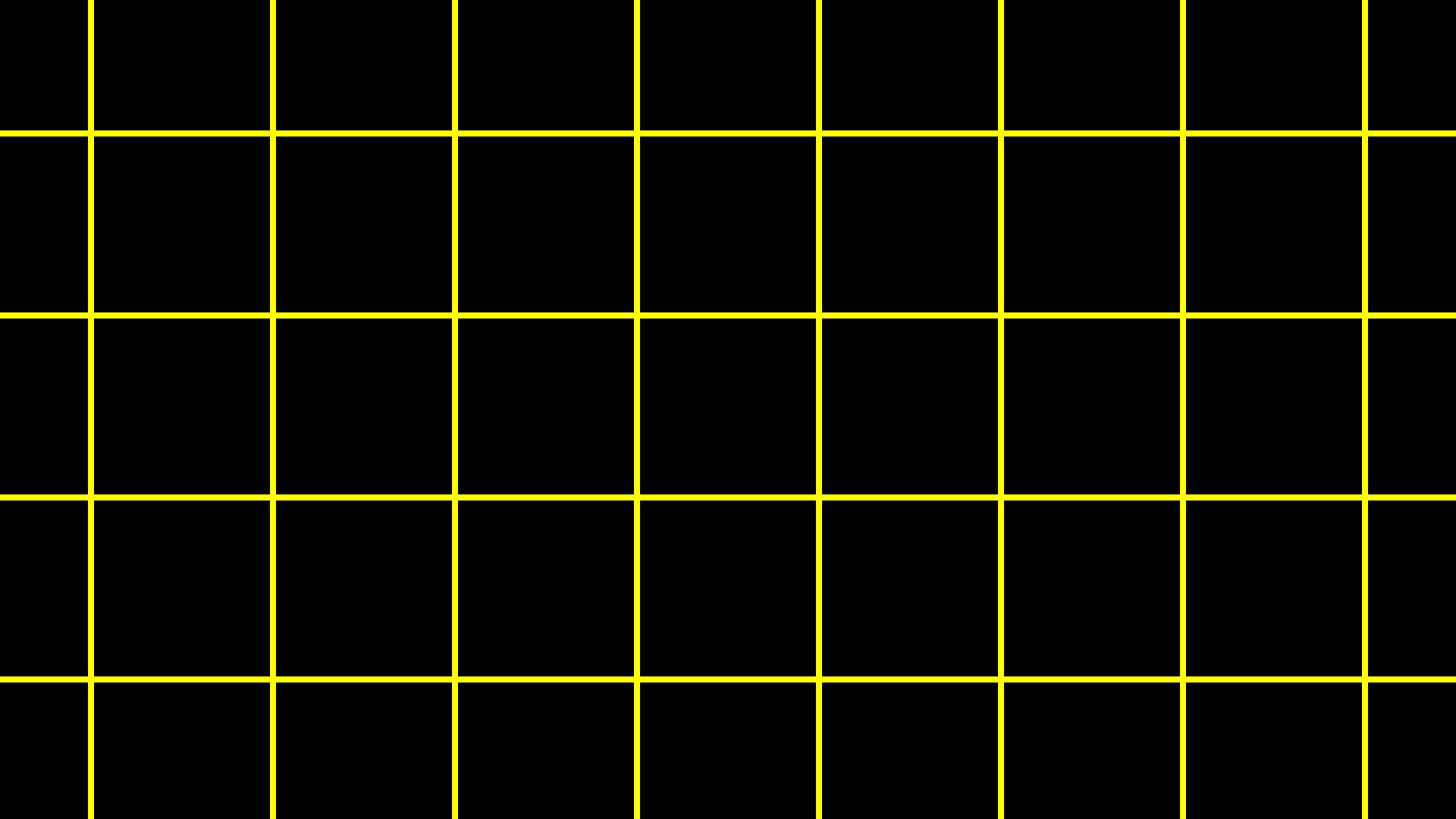
4G85

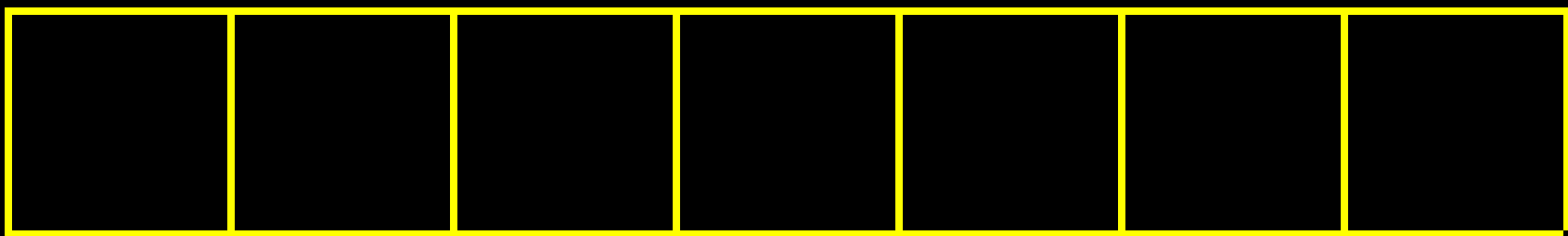


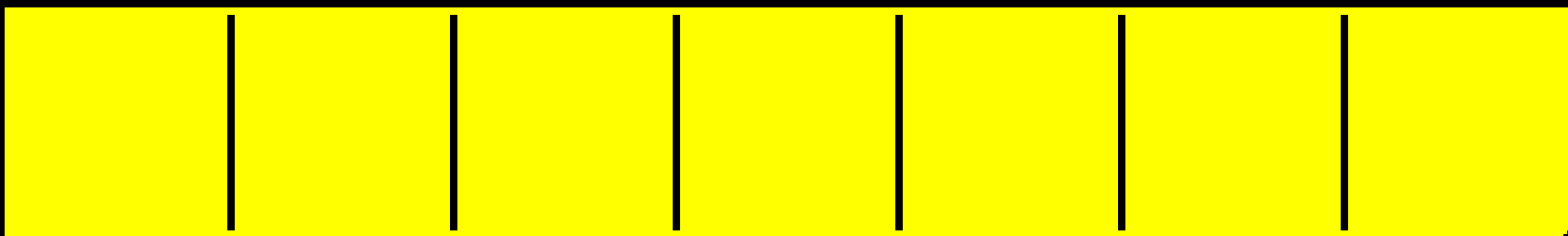
8BB12
D9HXT

4G85









linear search

For each door from left to right

 If number is behind door

 Answer is true

Answer is false

binary search

```
If number behind middle door
    Return true
Else if number < middle door
    Search left half
Else if number > middle door
    Search right half
```

If no doors

If number behind middle door

Return true

Else if number < middle door

Search left half

Else if number > middle door

Search right half

If no doors

 Return false

If number behind middle door

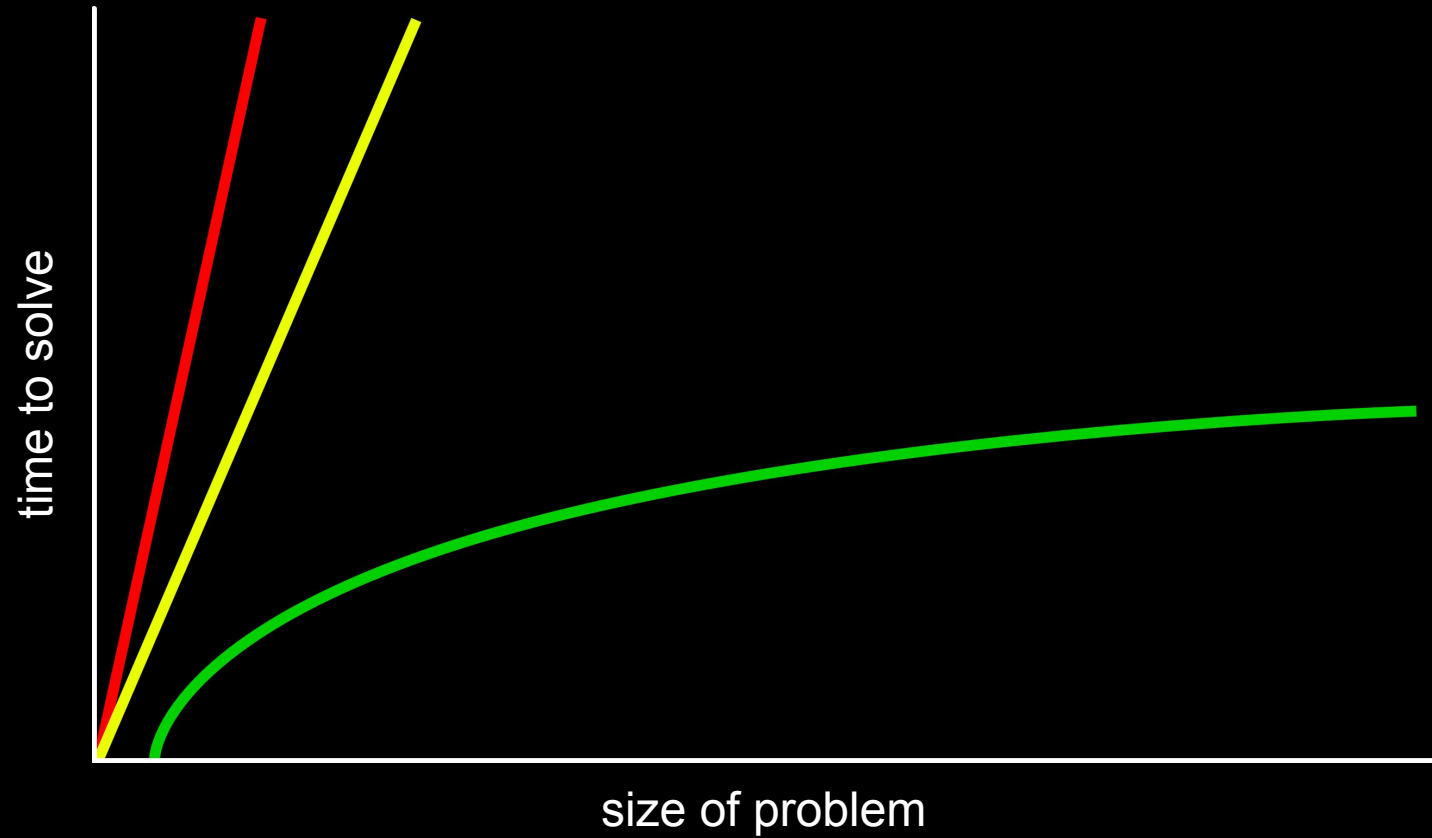
 Return true

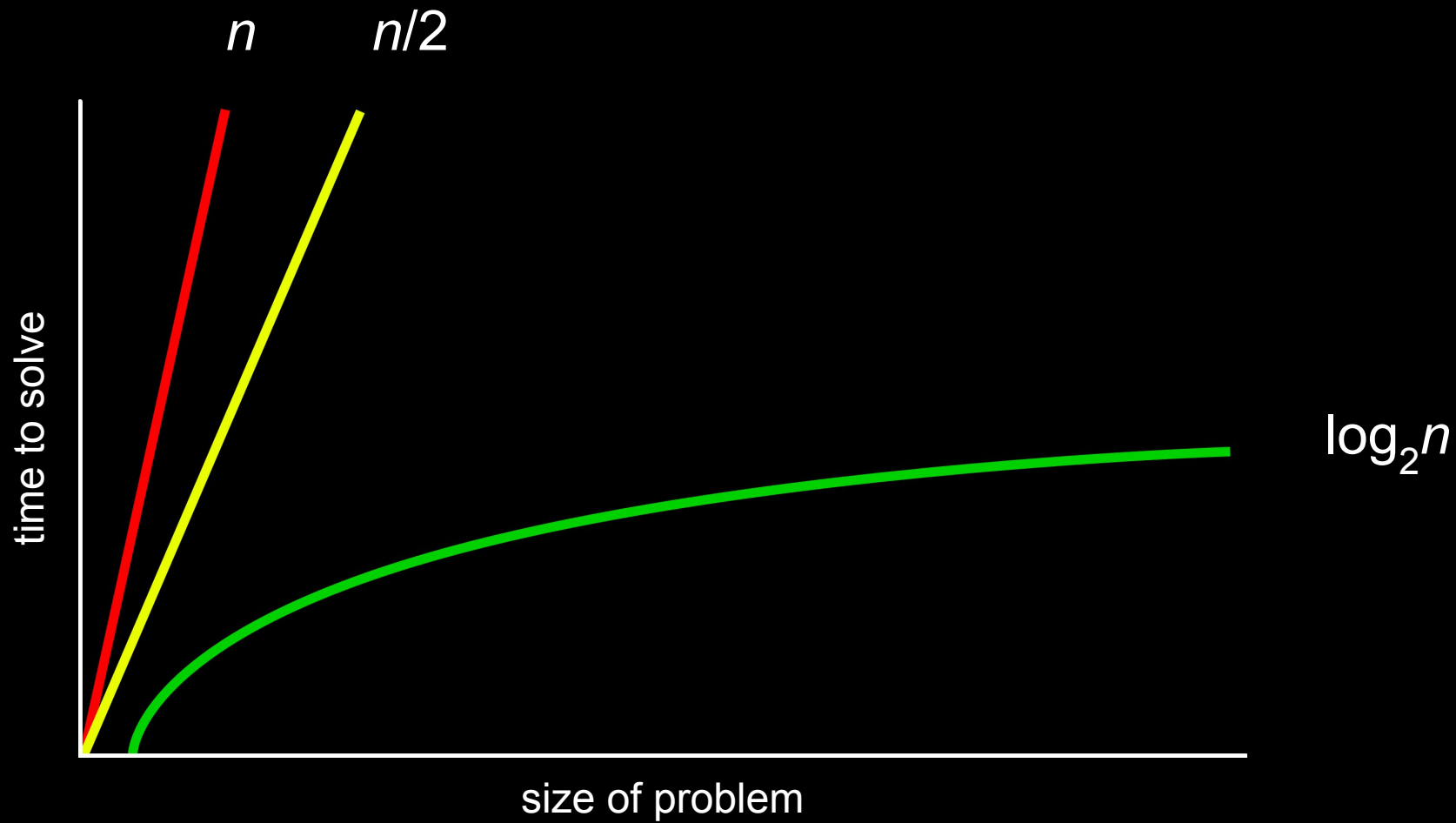
Else if number < middle door

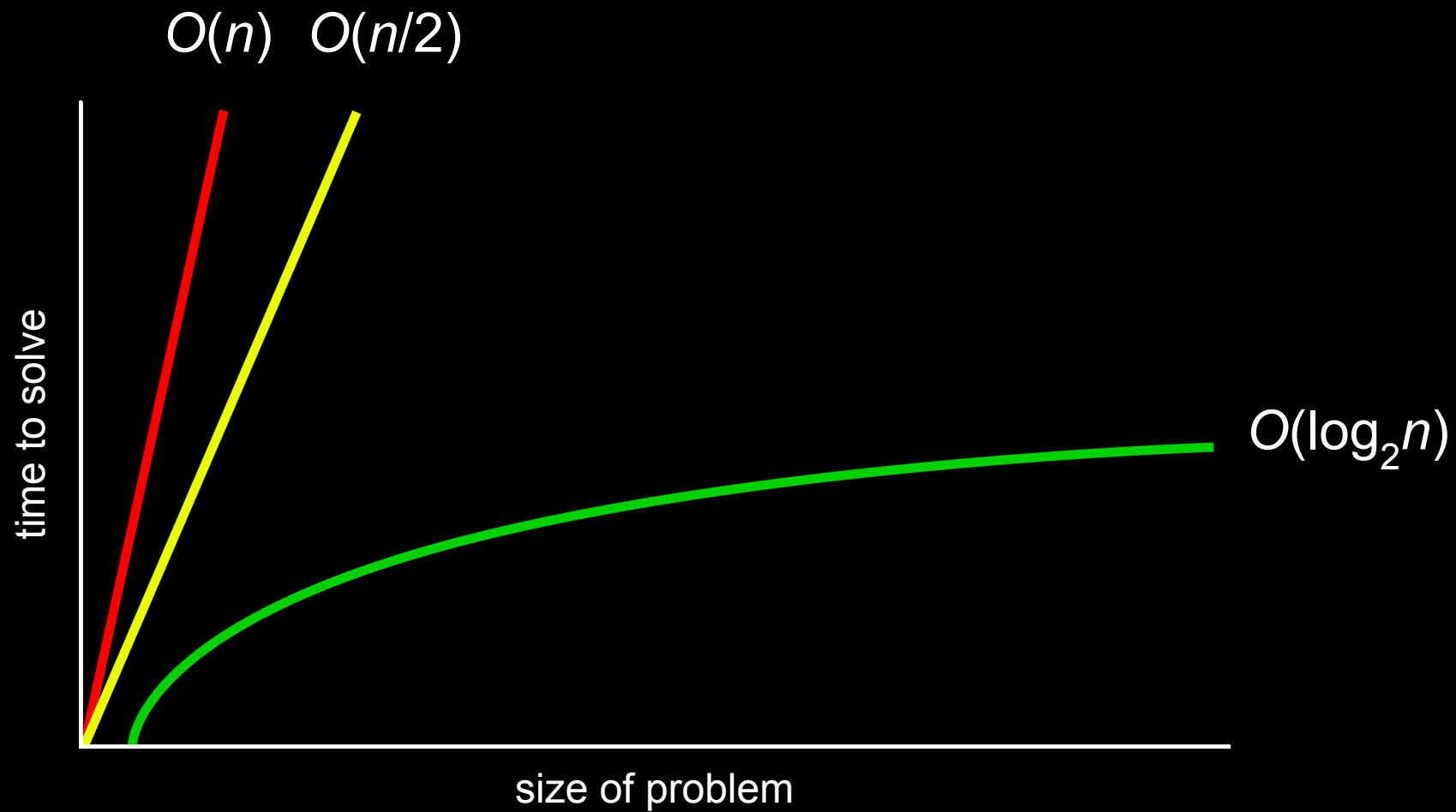
 Search left half

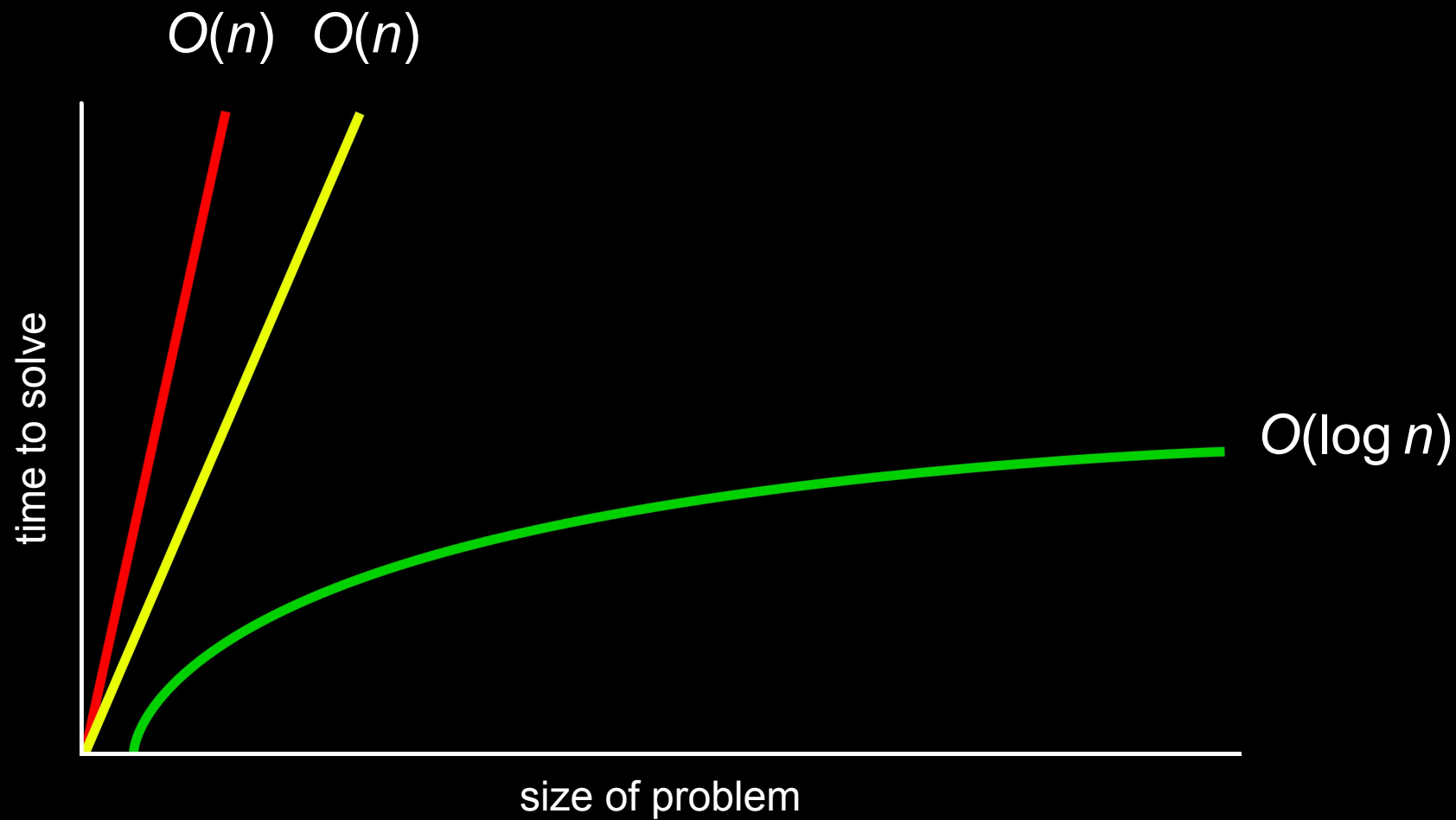
Else if number > middle door

 Search right half









$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

$O(n^2)$

$O(n \log n)$

$O(n)$ linear search

$O(\log n)$ binary search

$O(1)$

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

$\Omega(n^2)$

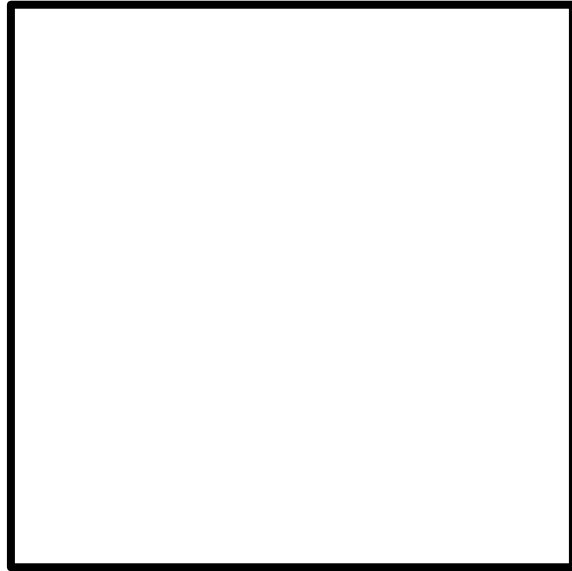
$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

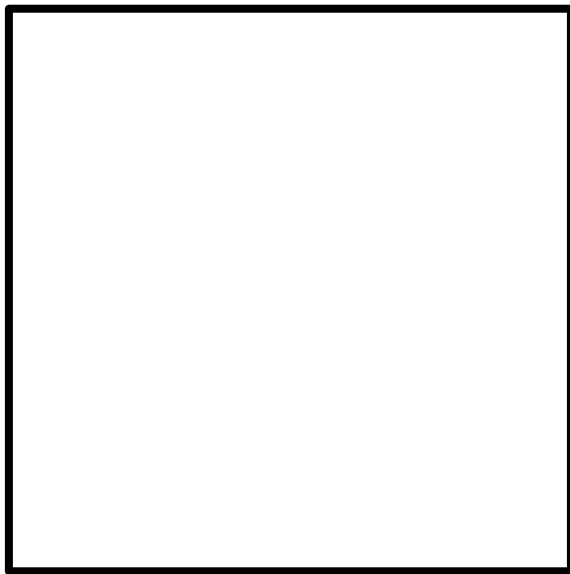
$\Omega(1)$ linear search, binary search

input →



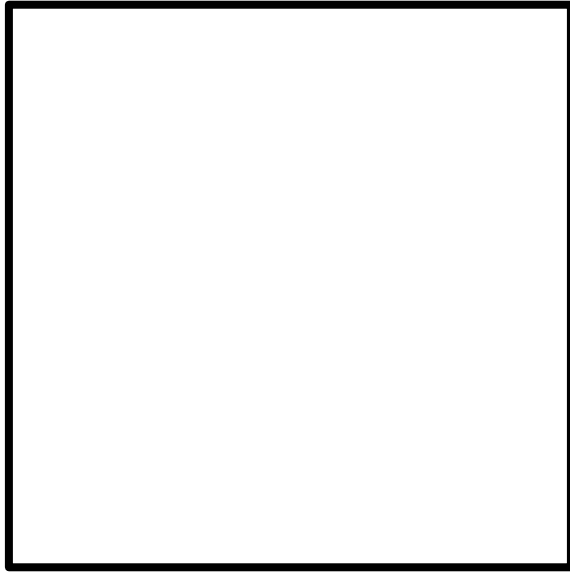
→ output

unsorted



output

unsorted



sorted

6 3 8 5 2 7 4 1

selection sort

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

$$n + (n - 1)$$

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

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

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

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

$$n(n + 1)/2$$

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

$$n(n + 1)/2$$

$$(n^2 + n)/2$$

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

$$n(n + 1)/2$$

$$(n^2 + n)/2$$

$$n^2/2 + n/2$$

$$O(n^2)$$

$O(n^2)$

selection sort

$\Omega(n^2)$ selection sort

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

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

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

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

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

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

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

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

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

$$O(n^2)$$

$O(n^2)$ bubble sort

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

$\Omega(n^2)$ bubble sort

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

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

Repeat until no swaps

For i from 0 to n-2

 If i'th and i+1'th elements out of order

 Swap them

$O(n^2)$ bubble sort

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$ bubble sort

$\Omega(\log n)$

$\Omega(1)$



recursion

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

```
1 Pick up phone book
2 Open to middle of phone book
3 Look at page
4 If person is on page
5     Call person
6 Else if person is earlier in book
7     Search left half of book
8 Else if person is later in book
9     Search right half of book
10 Else
11     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

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



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

2 4 5 7

1

3 6

8

2

4

5

7

1

3

6

8

2

4

5

7

1

3

6

8

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

1 2 3 4

5 7

6 8

7

6 8

1 2 3 4 5

7

8

1

2

3

4

5

6

1 2 3 4 5 6 7

8

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 |
|---|---|---|---|---|---|---|---|

$O(n^2)$

$O(n \log n)$ merge sort

$O(n)$

$O(\log n)$

$O(1)$

$\Omega(n^2)$

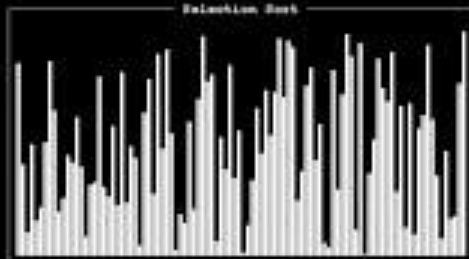
$\Omega(n \log n)$ merge sort

$\Omega(n)$

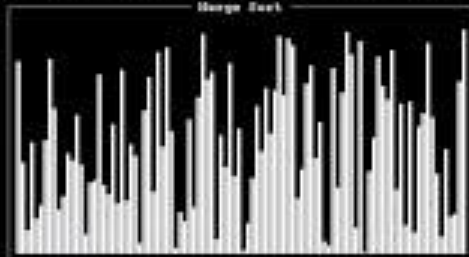
$\Omega(\log n)$

$\Omega(1)$

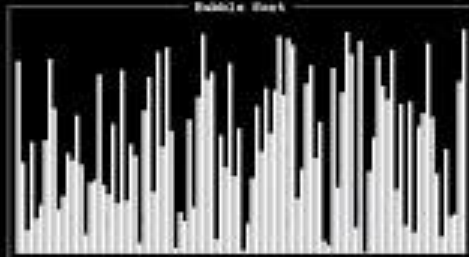
Examination Book



Home Book



Public Book



data structures

| | |
|--------------------|----------------------|
| <code>bool</code> | Boolean value |
| <code>float</code> | floating-point value |
| <code>int</code> | integer |
| <code>str</code> | string |
| <code>...</code> | |

dict

list

range

set

tuple

...





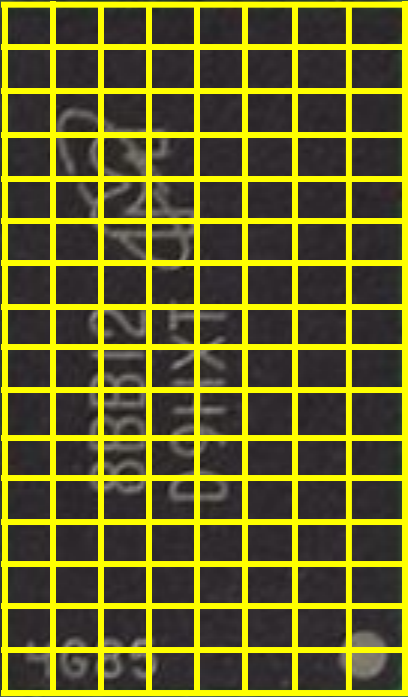
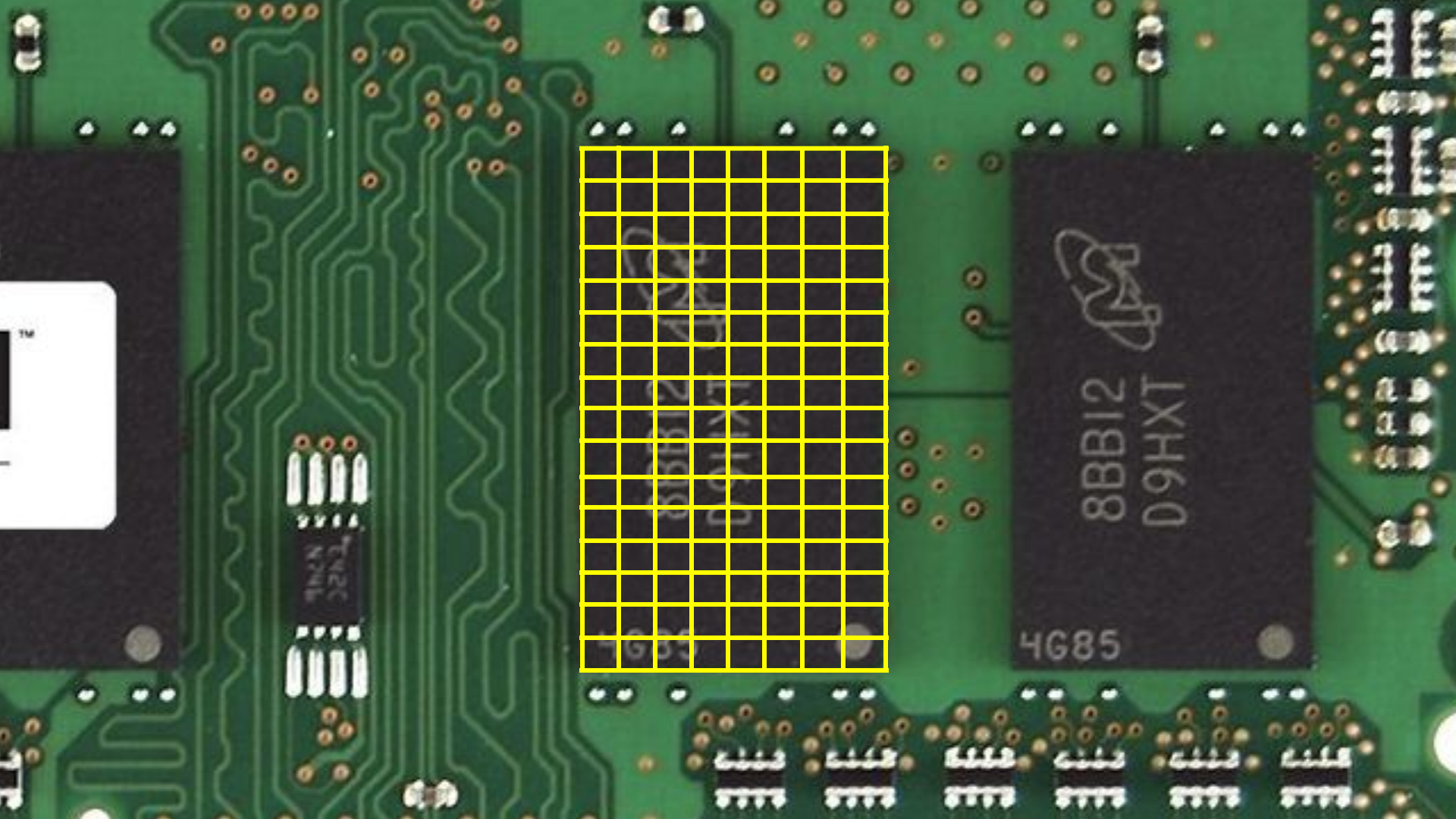
8BB12
D9HXT

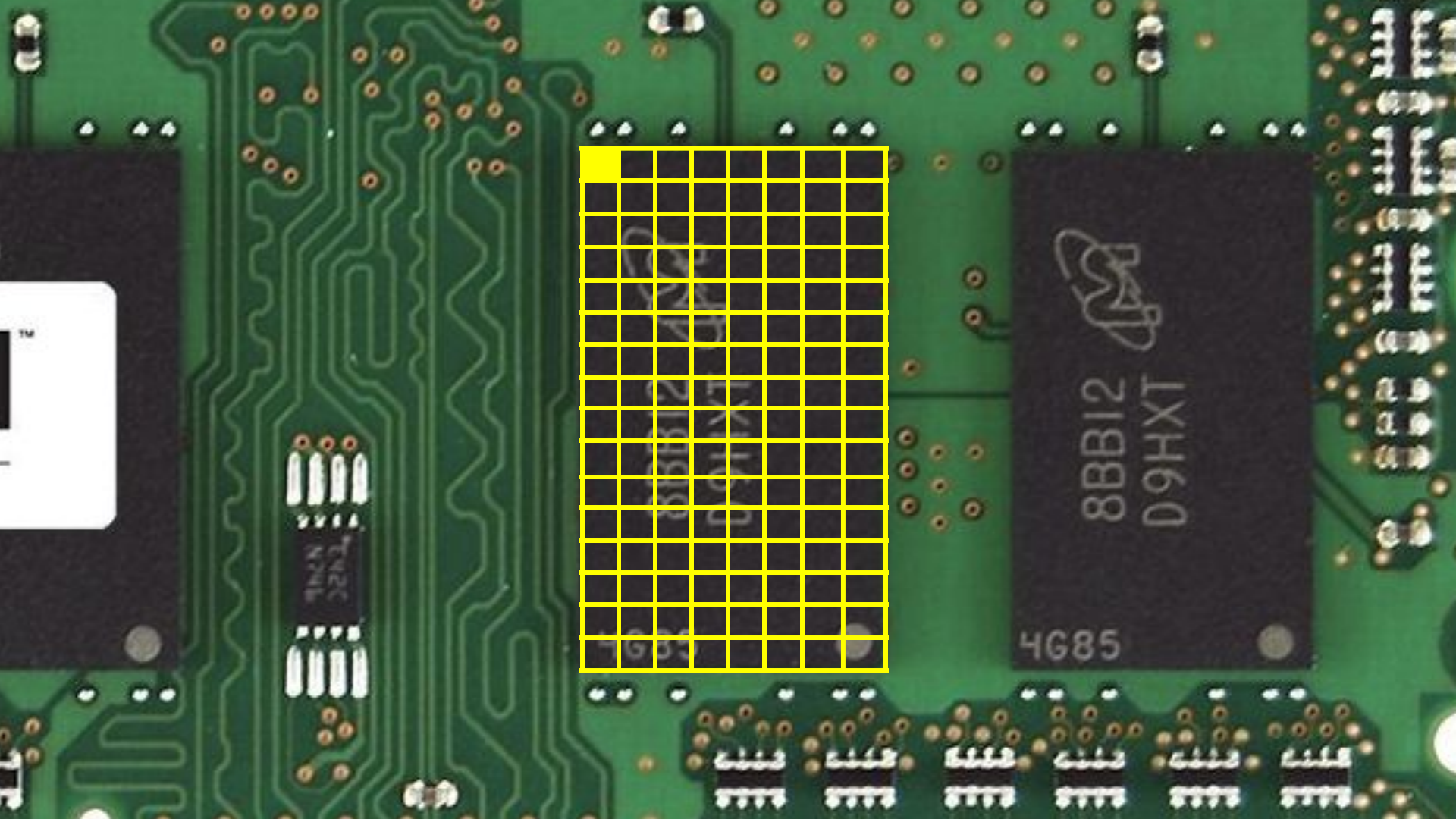
4G85



8BB12
D9HXT

4G85





| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

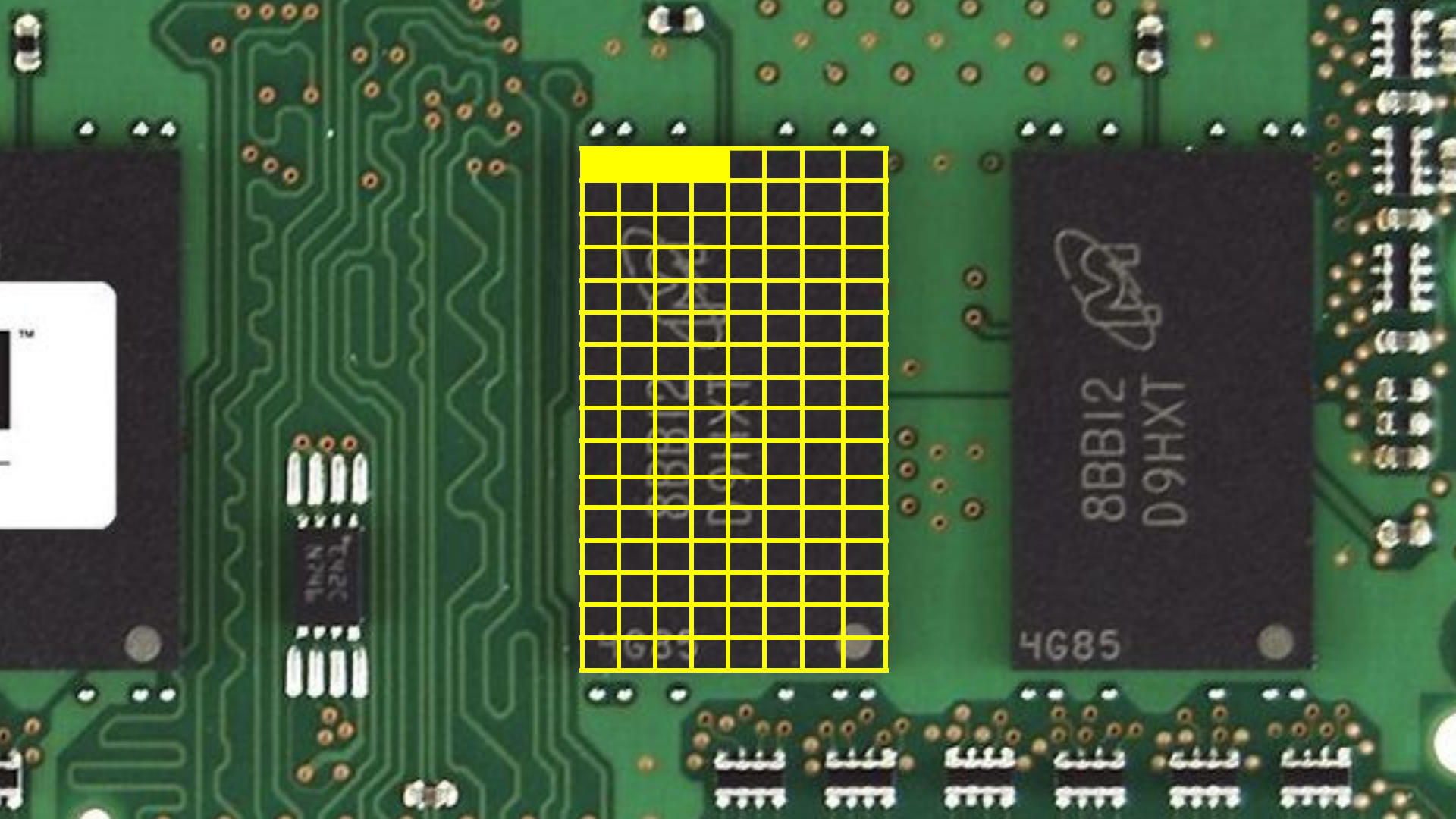


8BB12
D9HXT

4G85

74VHC245

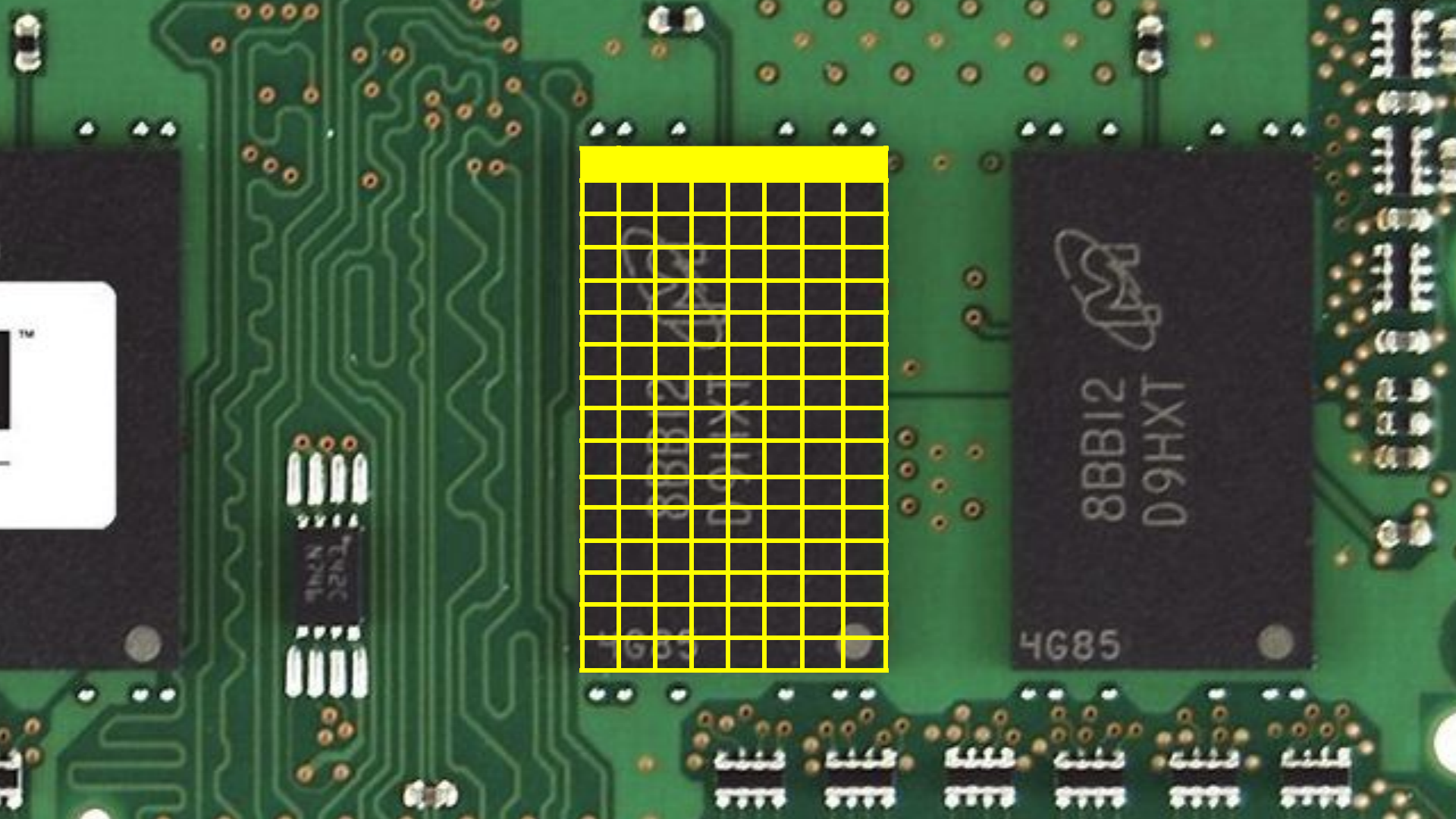
8BB12
D9HXT



| | | | | | | | | | |
|--|--|--|--|--|--|--|--|--|--|
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

8BB12
D9HXT
4G85

8BB12
D9HXT



| | | | | | | | | | | | |
|--------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| [Yellow grid overlaying a component] | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 |
| 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 |
| 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 |
| 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 |
| 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

8BB12
D9HXT
4G85

arrays





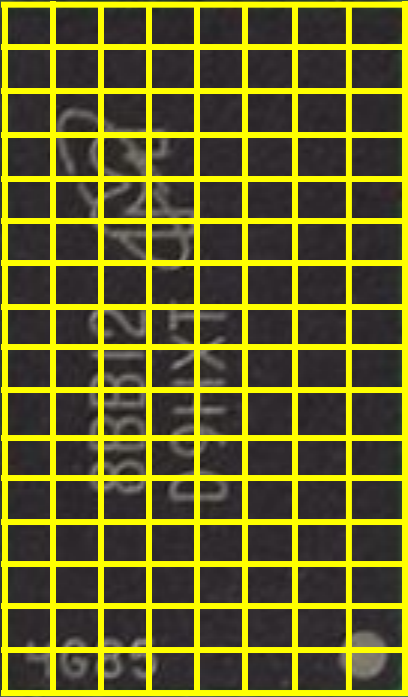
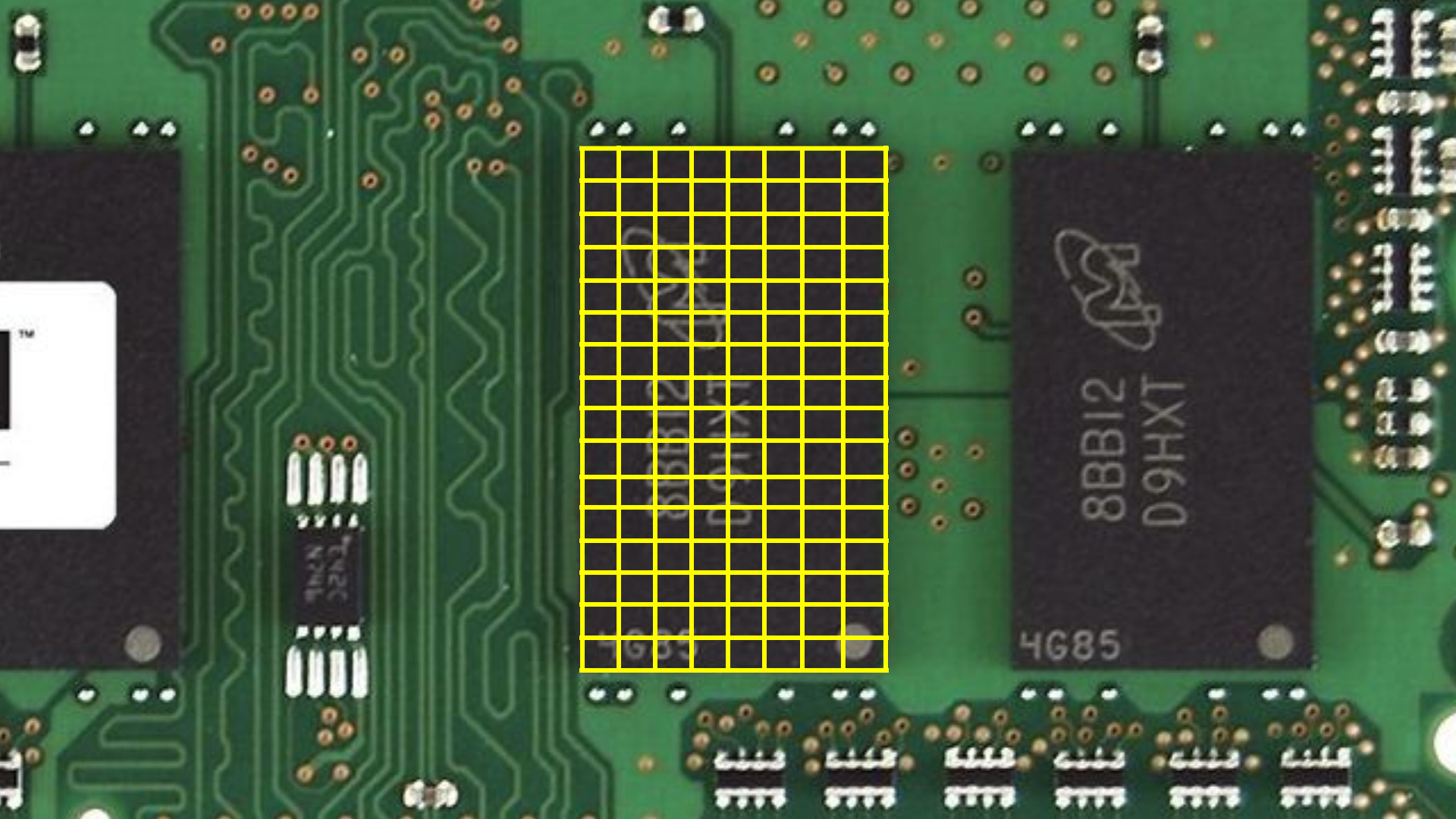
8BB12
D9HXT

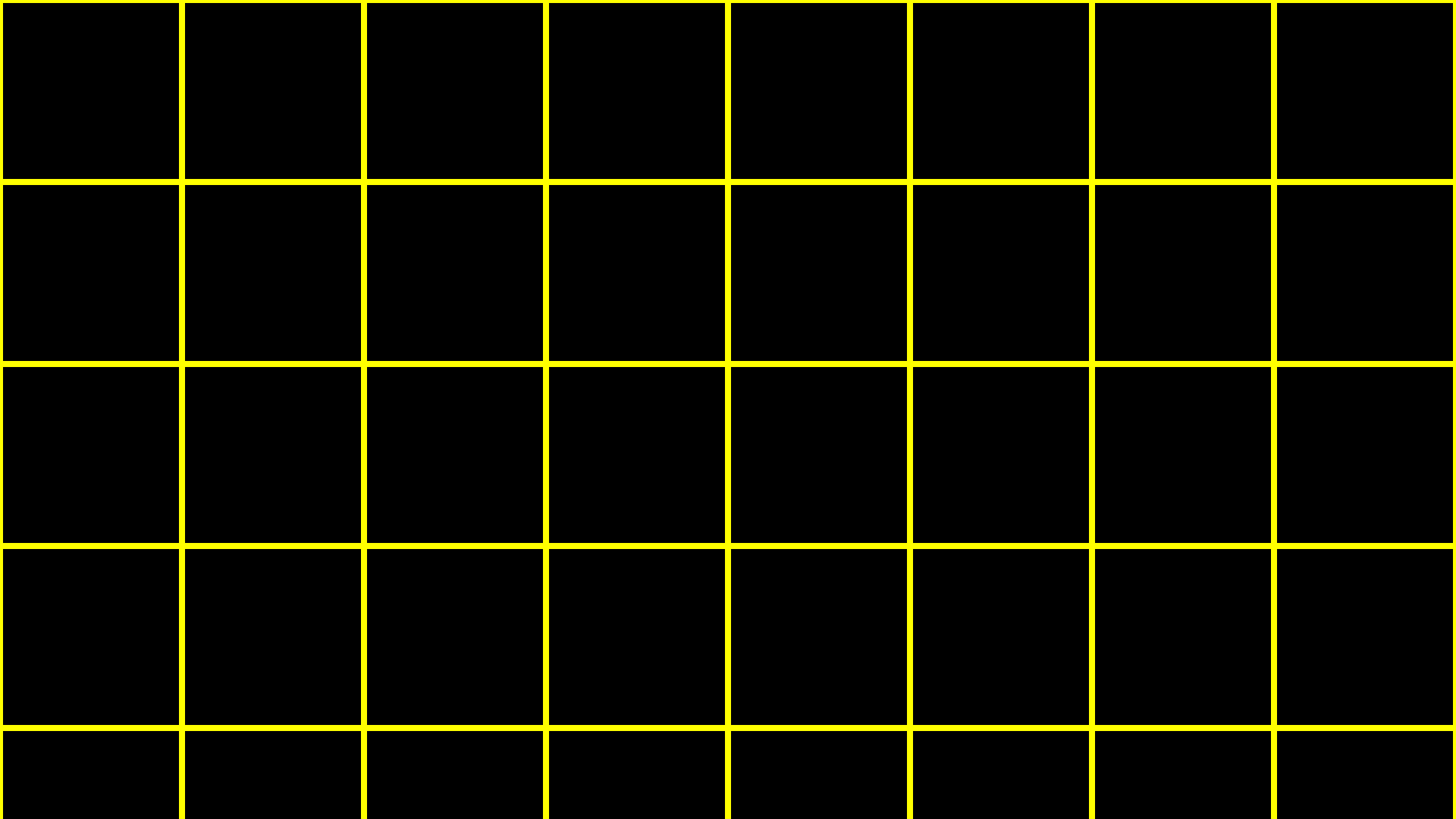
4G85

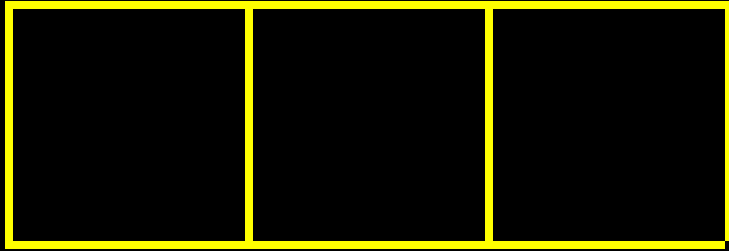


8BB12
D9HXT

4G85





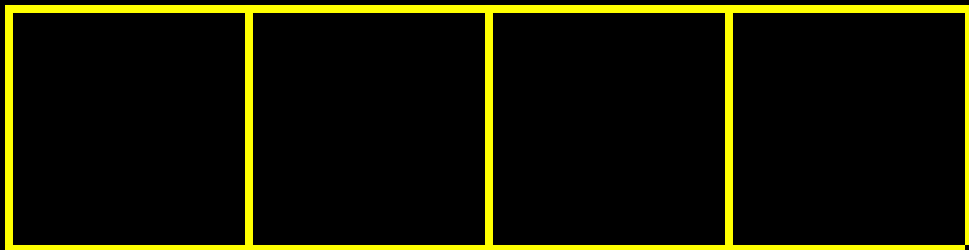
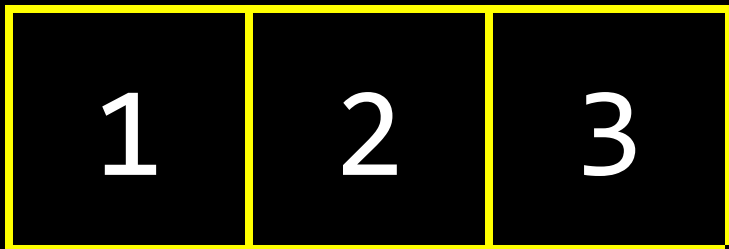


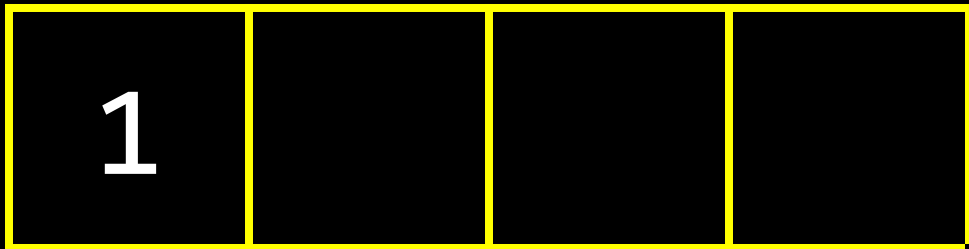
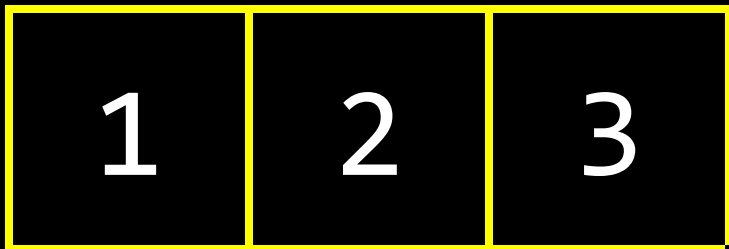
| | | |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|

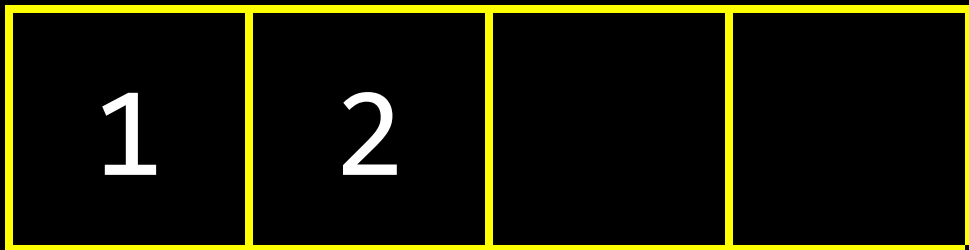
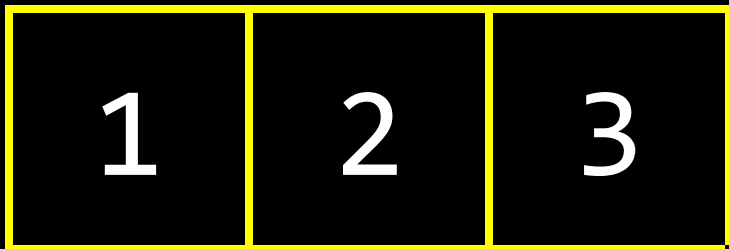
| | | | |
|---|---|---|--|
| 1 | 2 | 3 | |
|---|---|---|--|

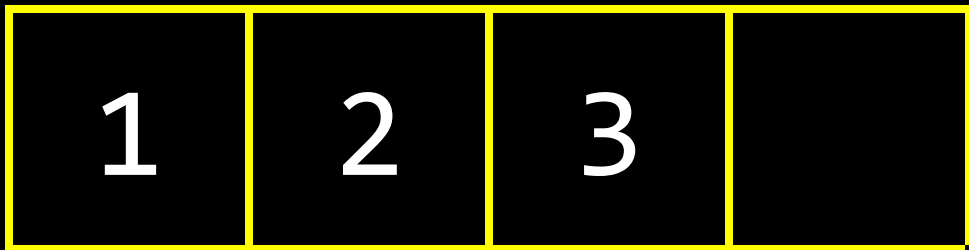
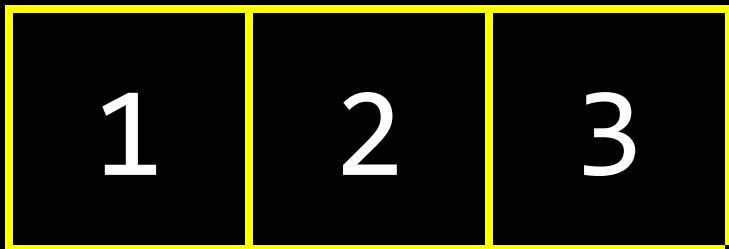
| | | | | | | | |
|--|---|---|---|--|--|--|--|
| | | | | | | | |
| | 1 | 2 | 3 | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

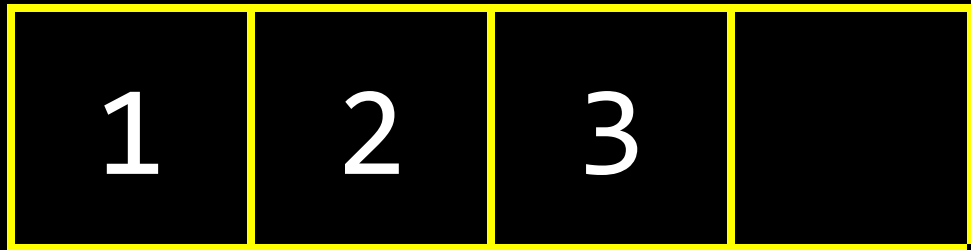
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ? | ? | ? | ? | ? | ? | ? | ? |
| ? | 1 | 2 | 3 | ? | ? | ? | ? |
| ? | ? | ? | ? | ? | ? | ? | ? |
| ? | ? | ? | | | | | |
| | | | | | | | |

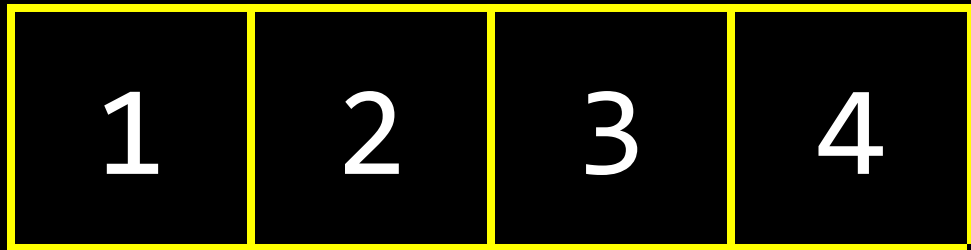












$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

$O(n^2)$

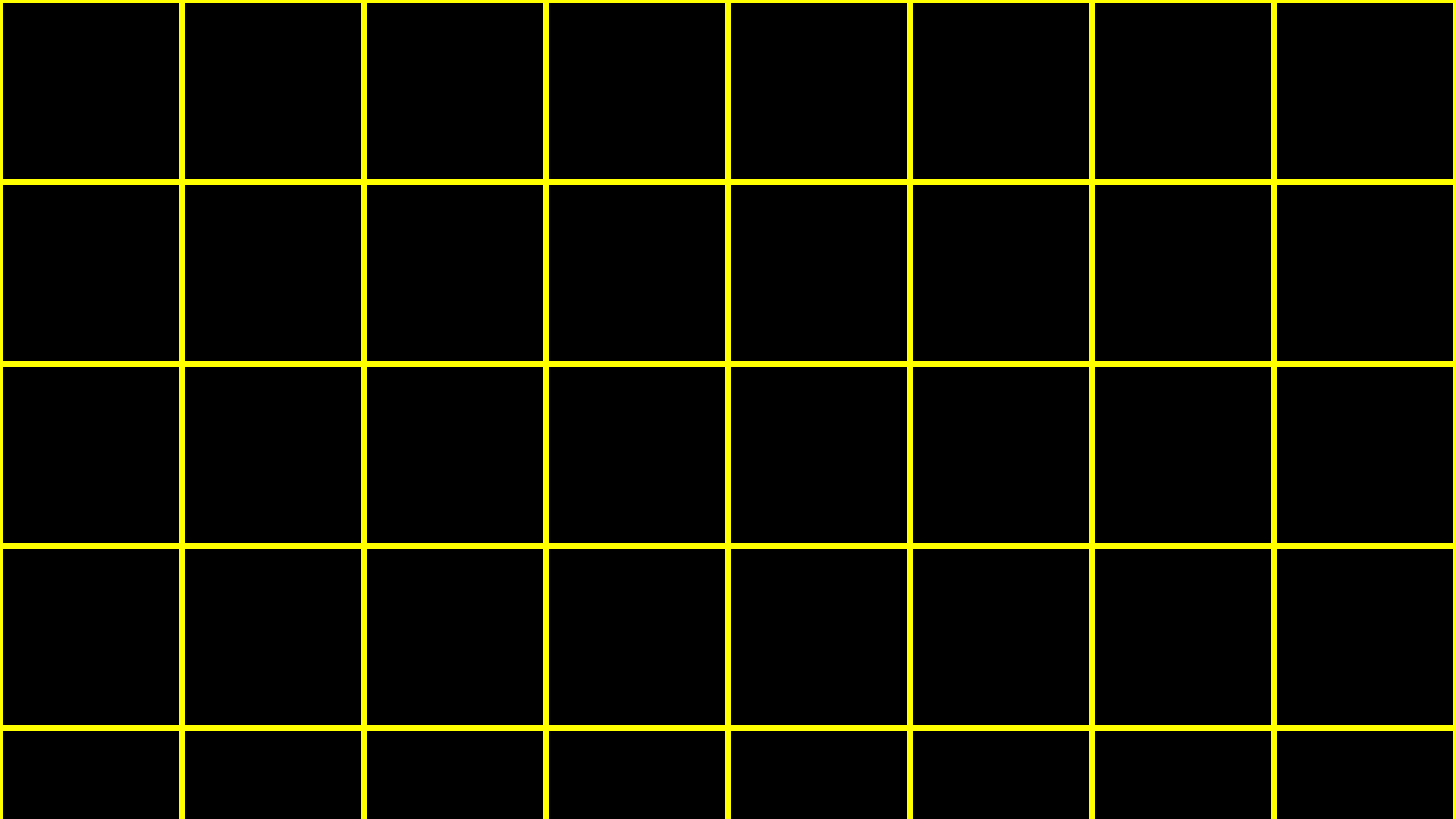
$O(n \log n)$

$O(n)$ insert

$O(\log n)$ search

$O(1)$

linked lists



1

0x123

1

0x123

2

0x456

1

0x123

2

0x456

3

0x789

1

0x123

2

0x456

3

0x789

1

0x123

0x456

2

0x456

3

0x789

1

0x123

0x456

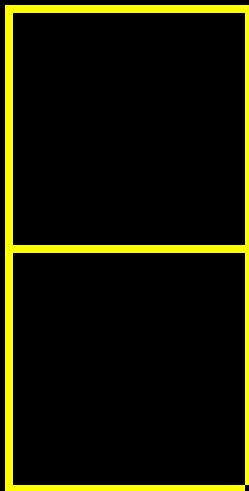
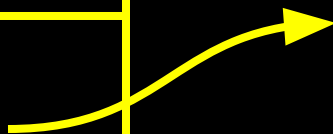
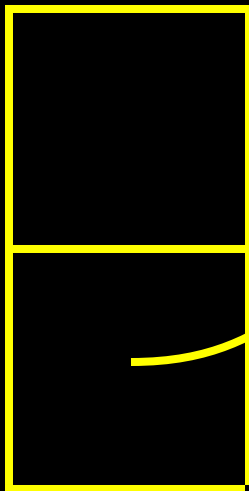
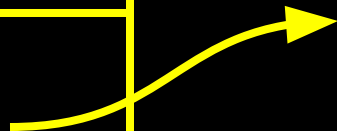
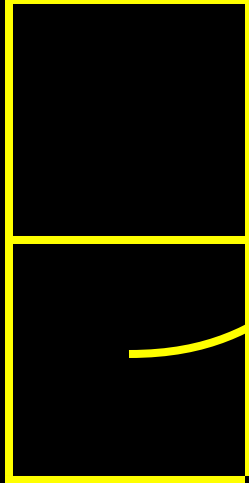
2

0x456

0x789

3

0x789



dict

list

range

set

tuple

...

trees

binary search trees

1

2

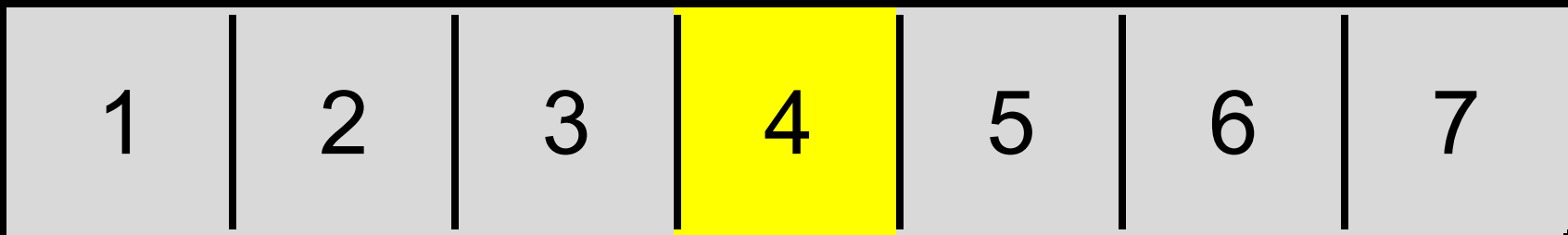
3

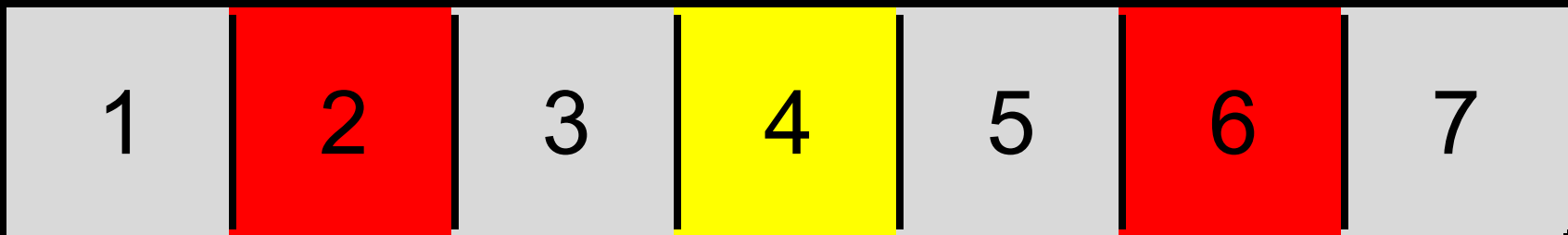
4

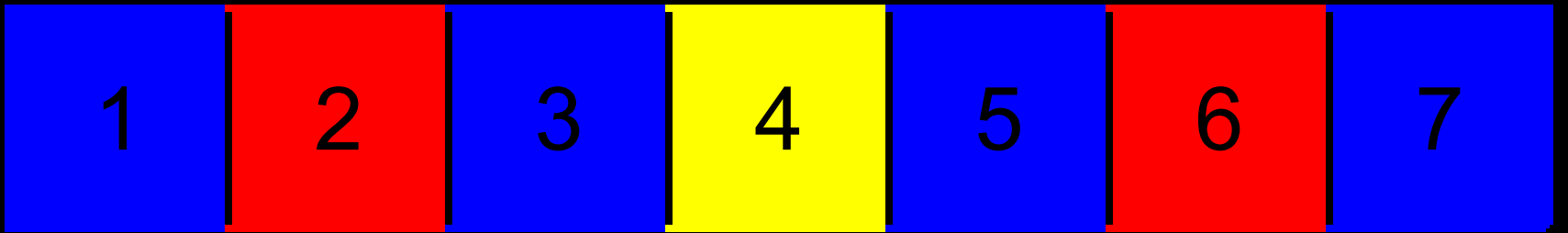
5

6

7







4

2

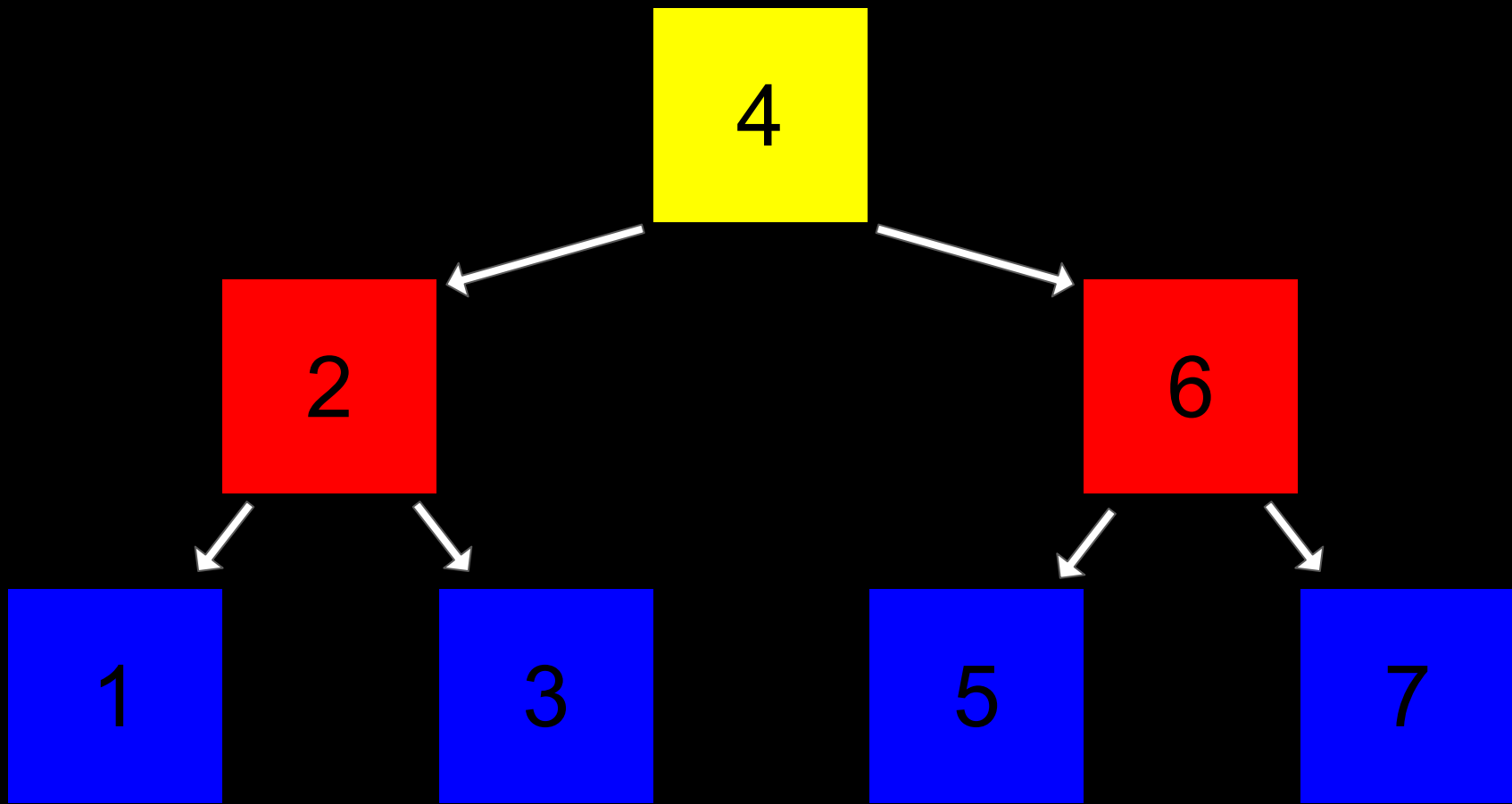
6

1

3

5

7



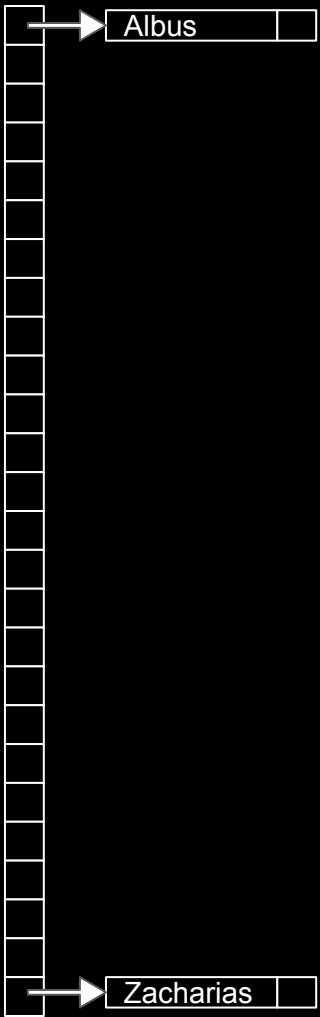
hash tables

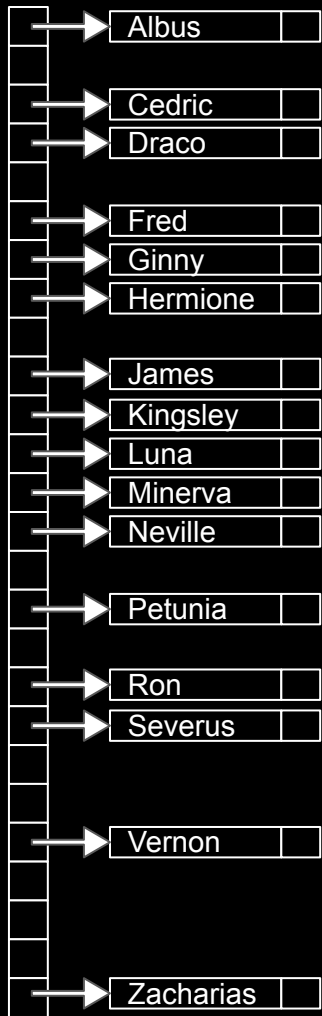


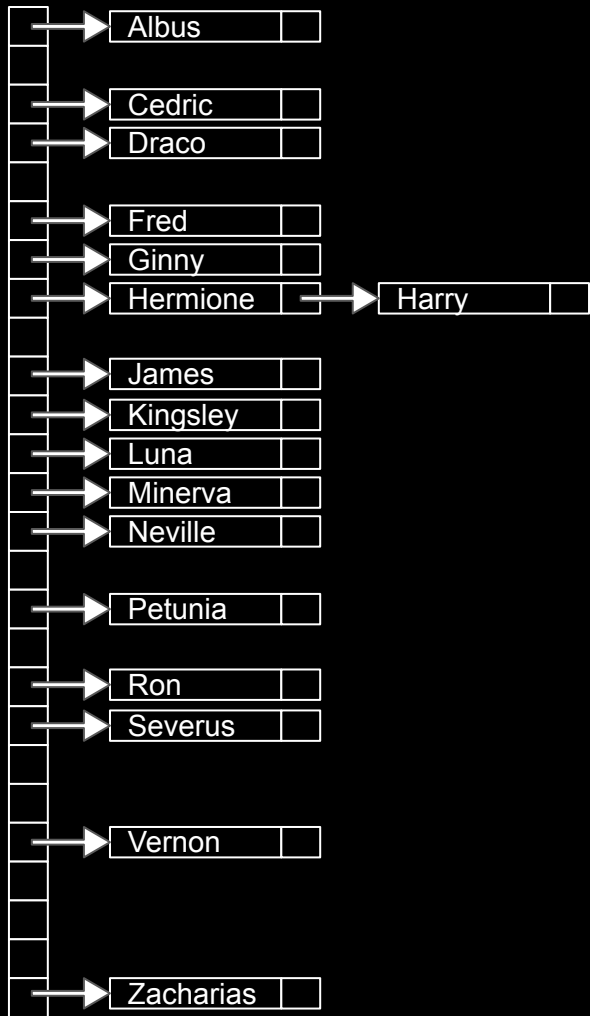
| | |
|----|--|
| 0 | |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |
| 11 | |
| 12 | |
| 13 | |
| 14 | |
| 15 | |
| 16 | |
| 17 | |
| 18 | |
| 19 | |
| 20 | |
| 21 | |
| 22 | |
| 23 | |
| 24 | |
| 25 | |

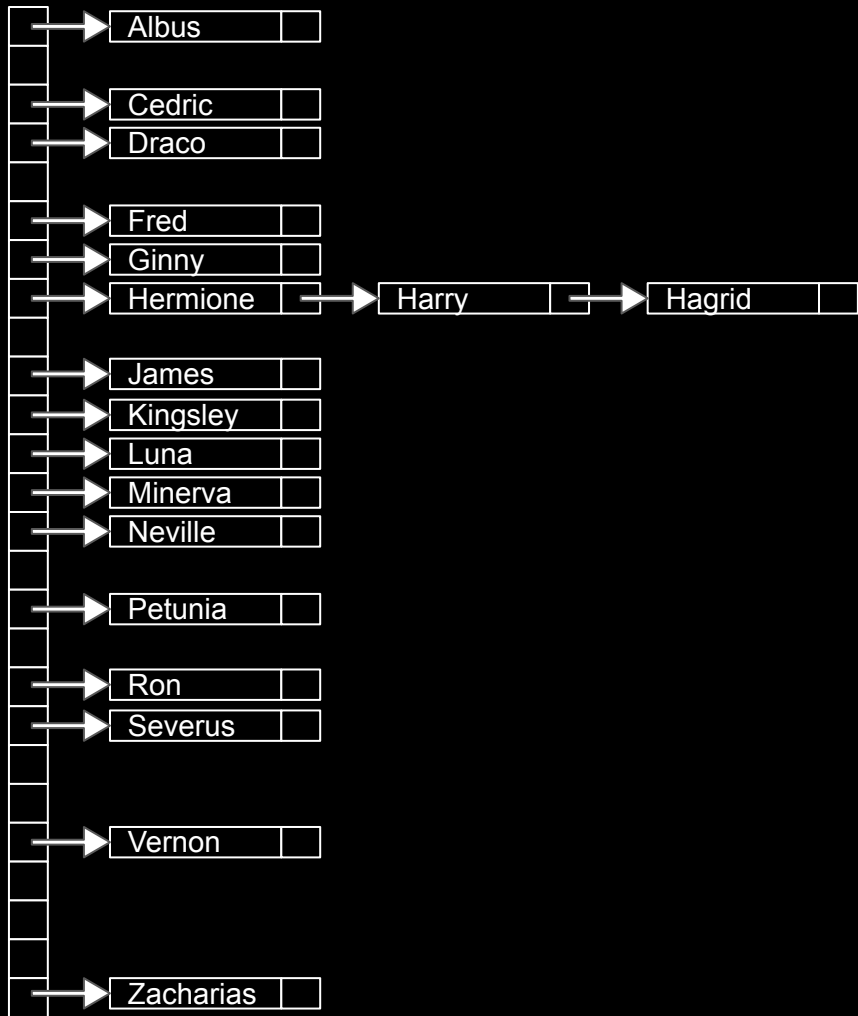
| | |
|---|--|
| A | |
| B | |
| C | |
| D | |
| E | |
| F | |
| G | |
| H | |
| I | |
| J | |
| K | |
| L | |
| M | |
| N | |
| O | |
| P | |
| Q | |
| R | |
| S | |
| T | |
| U | |
| V | |
| W | |
| X | |
| Y | |
| Z | |

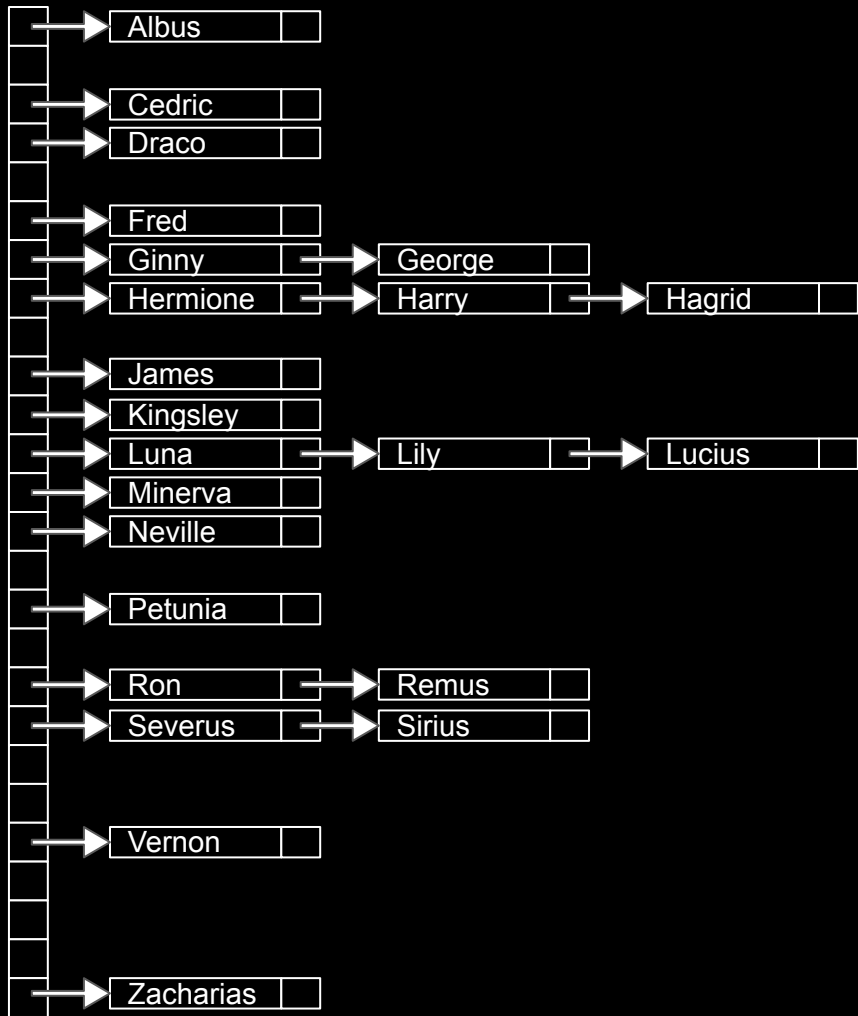




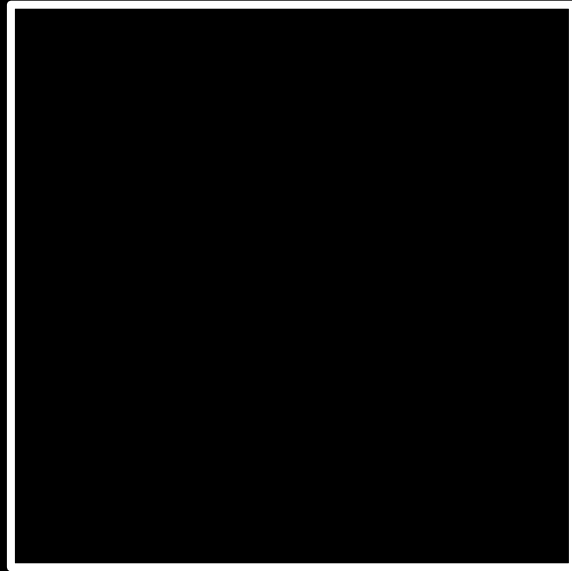








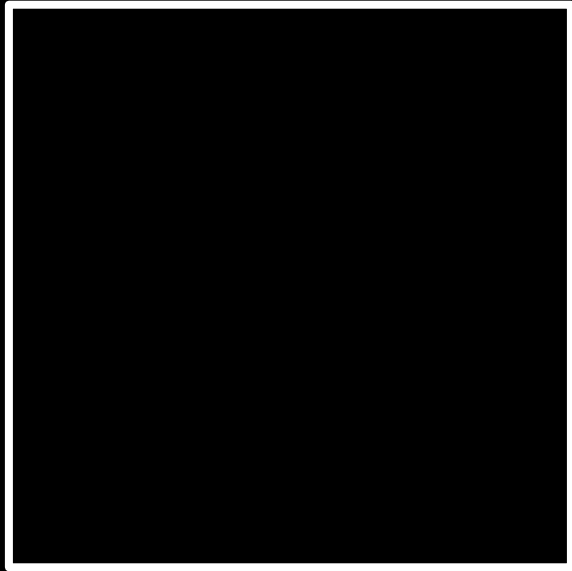
input →



→ output

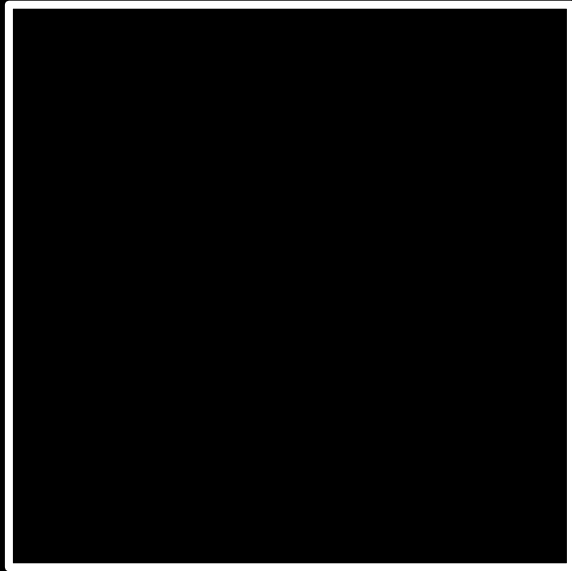
hash function

Albus →

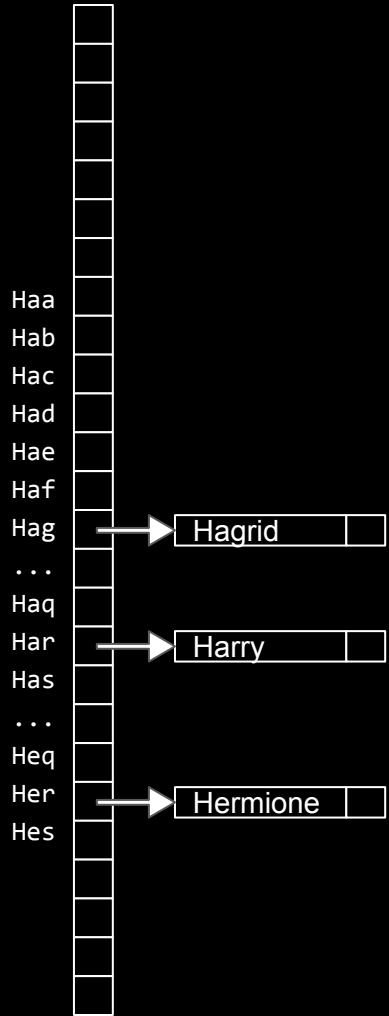


→ 0

Zacharias →



→ 25



$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

$O(n^2)$

$O(n \log n)$

$O(n)$ search

$O(\log n)$

$O(1)$

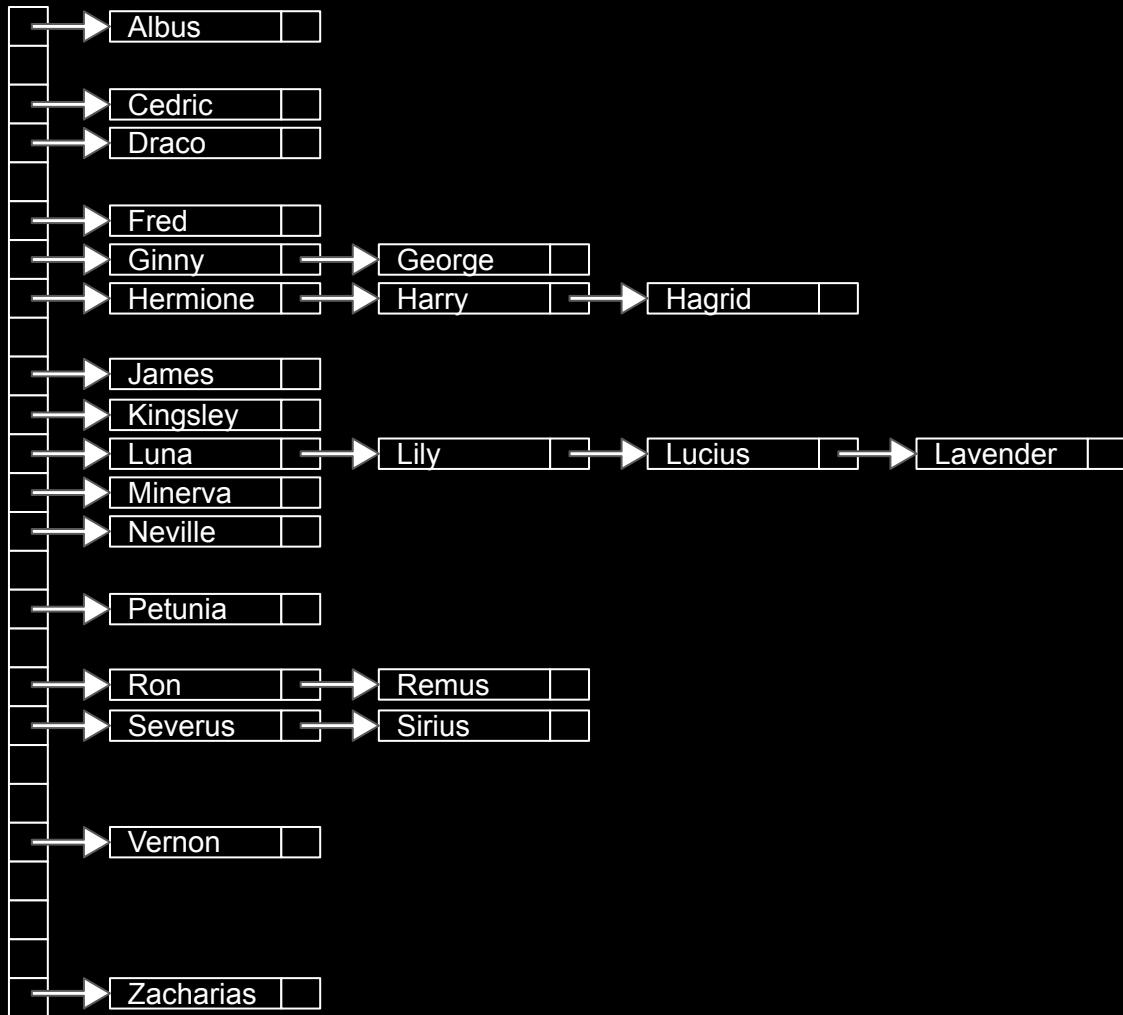
$O(n^2)$

$O(n \log n)$

$O(n)$ search

$O(\log n)$

$O(1)$ insert



dictionaries

dict

list

range

set

tuple

...

PICK ME UP



B C D E F G H I



K L M N O P Q R



T U- V W X Y Z



CHOKIN
BOOK IS GREAT
VISION STOPS RESPONSE
+

5 5 5

Lab 0

Assignment 2

Office Hours

CS50 for JDs

Algorithms, Data Structures