This is CS50.

cs50.brianyu.me

Week 3

- Searching (Linear, Binary)
- Sorting (Bubble, Selection)
- Big O
- Structs
- Recursion
- Merge Sort

What questions do you have?

Today

Search and Sort

Recursion and Structs

Lab

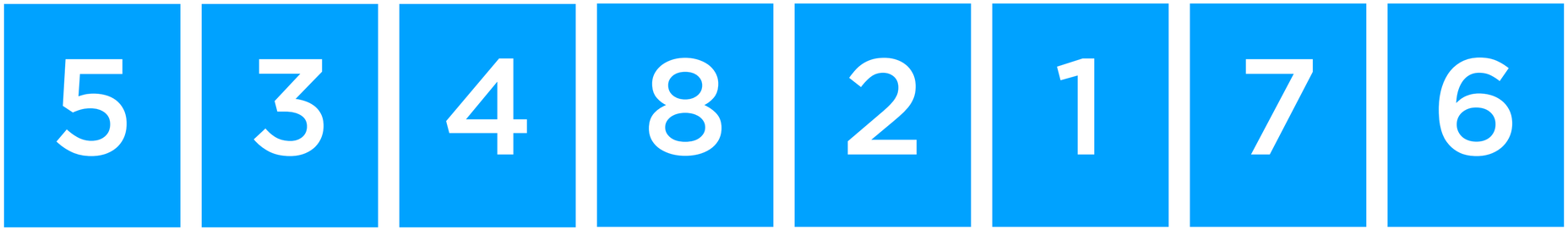
Part One

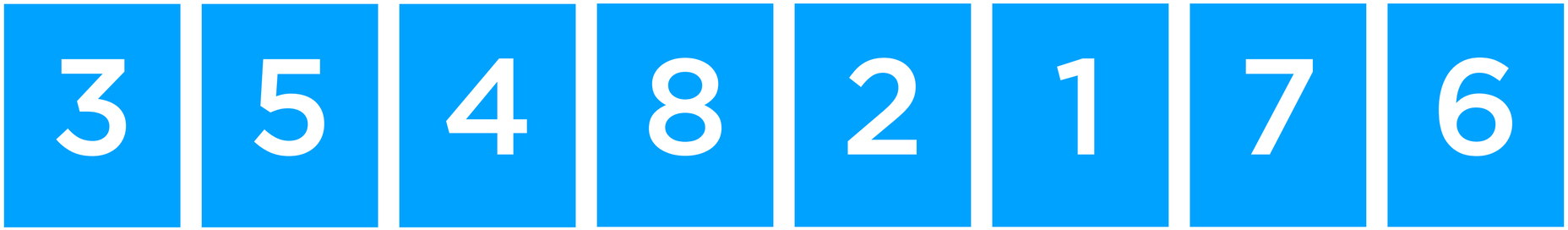
Search and Sort

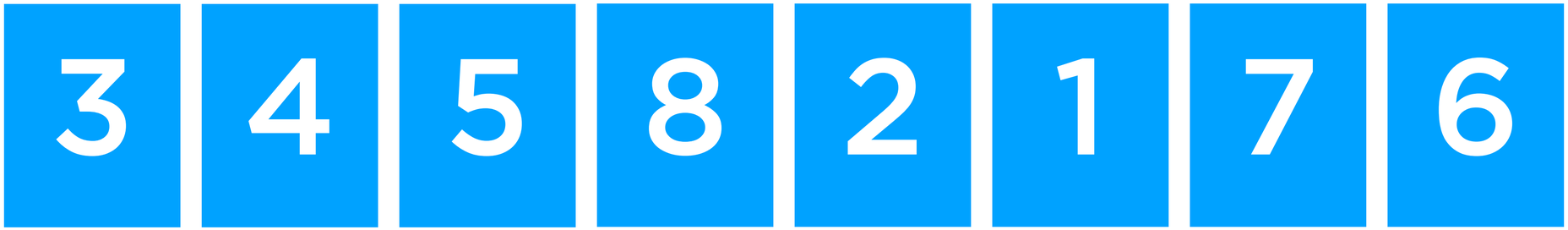
Linear Search

Binary Search

Bubble Sort


5 3 4 8 2 1 7 6


3 5 4 8 2 1 7 6


3 4 5 8 2 1 7 6


3 4 5 2 8 1 7 6


3 4 5 2 1 8 7 6


3 4 5 2 1 7 8 6


3 4 5 2 1 7 6 8


3
4
5
2
1
7
6
8


3 4 2 5 1 7 6 8


3 4 2 1 5 7 6 8


3
4
2
1
5
6
7
8


3 4 2 1 5 6 7 8


3
2
4
1
5
6
7
8


3
2
1
4
5
6
7
8


3 2 1 4 5 6 7 8


2 3 1 4 5 6 7 8


2 1 3 4 5 6 7 8

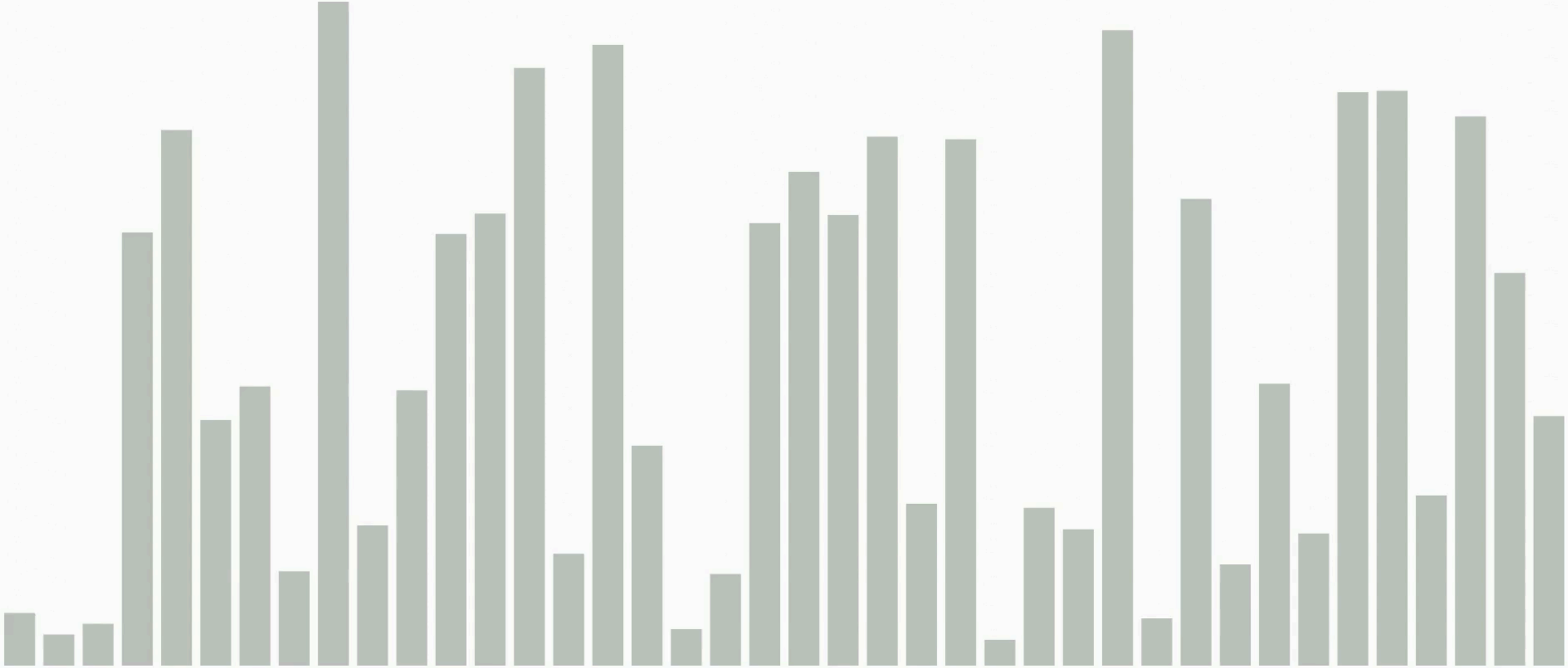

2 1 3 4 5 6 7 8


1
2
3
4
5
6
7
8


1
2
3
4
5
6
7
8


1
2
3
4
5
6
7
8

Bubble Sort



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

cs50.brianyu.me/sorting.c

Exercise

Complete **sorting.c** such that it sorts integers using Bubble Sort.

cs50.brianyu.me/sorting.c

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

Bubble Sort

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

Bubble Sort

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

```
for (int i = 0; i < n - 1; i++)
{









}
```

Bubble Sort

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

```
for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - 1; j++)
    {









    }
}
```

Bubble Sort

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

```
for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - 1; j++)
    {
        if (values[j] > values[j + 1])
        {



        }
    }
}
```

Bubble Sort

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

```
for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - 1; j++)
    {
        if (values[j] > values[j + 1])
        {
            int temp = values[j];



        }
    }
}
```

Bubble Sort

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

```
for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - 1; j++)
    {
        if (values[j] > values[j + 1])
        {
            int temp = values[j];
            values[j] = values[j + 1];

        }
    }
}
```

Bubble Sort

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

```
for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - 1; j++)
    {
        if (values[j] > values[j + 1])
        {
            int temp = values[j];
            values[j] = values[j + 1];
            values[j + 1] = temp;
        }
    }
}
```

Bubble Sort

```
for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - 1; j++)
    {
        if (values[j] > values[j + 1])
        {
            int temp = values[j];
            values[j] = values[j + 1];
            values[j + 1] = temp;
        }
    }
}
```

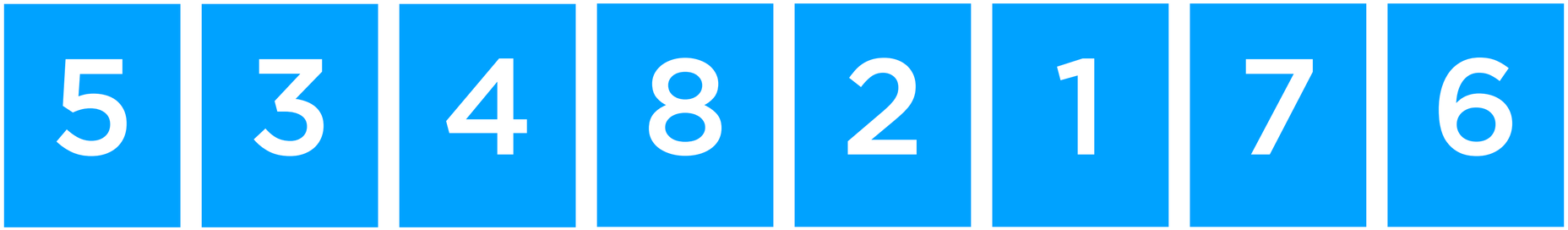
Bubble Sort

```
for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - 1 - i; j++)
    {
        if (values[j] > values[j + 1])
        {
            int temp = values[j];
            values[j] = values[j + 1];
            values[j + 1] = temp;
        }
    }
}
```

Bubble Sort

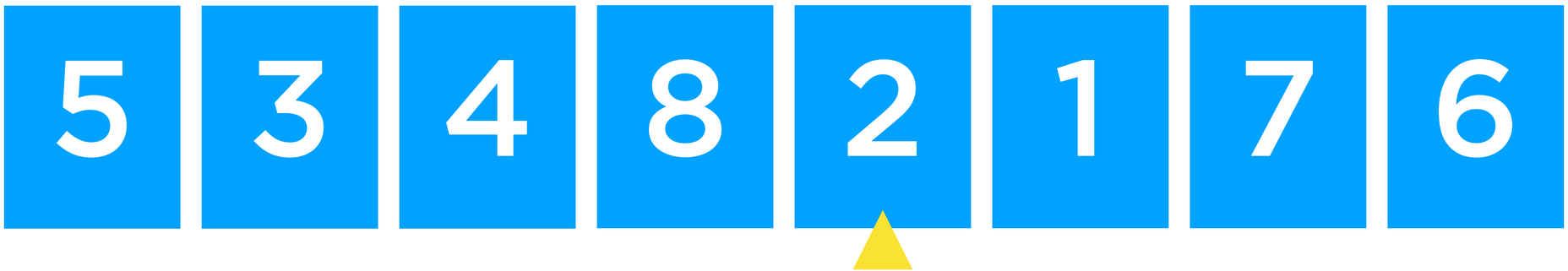
```
for (int i = 0; i < n - 1; i++)
{
    bool swaps = false;
    for (int j = 0; j < n - 1 - i; j++)
    {
        if (values[j] > values[j + 1])
        {
            int temp = values[j];
            values[j] = values[j + 1];
            values[j + 1] = temp;
            swaps = true;
        }
    }
    if (swaps == false)
        break;
}
```

Selection Sort

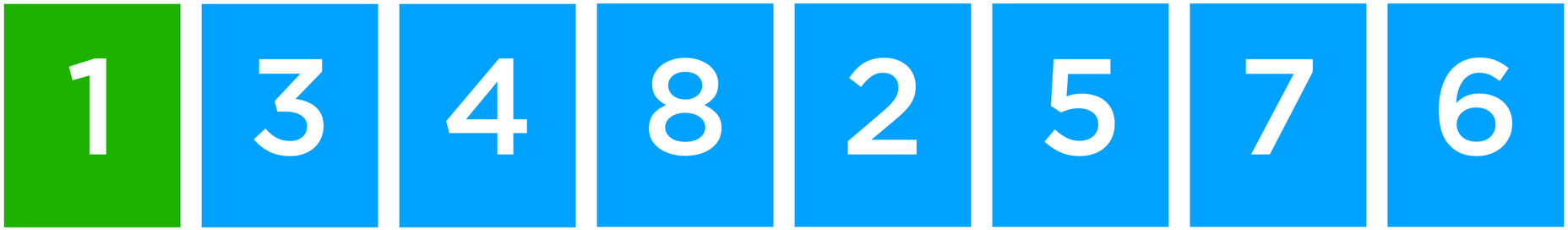

5 3 4 8 2 1 7 6


5 3 4 8 2 1 7 6

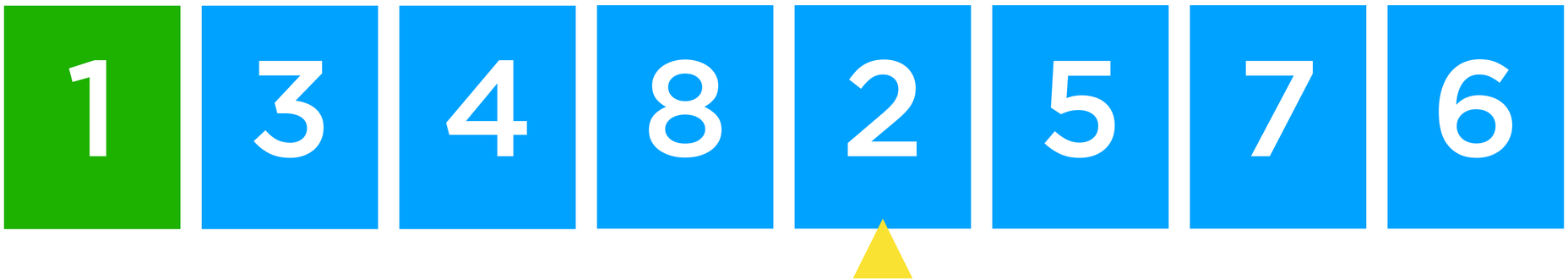

5 3 4 8 2 1 7 6


5 3 4 8 2 1 7 6


5 3 4 8 2 1 7 6

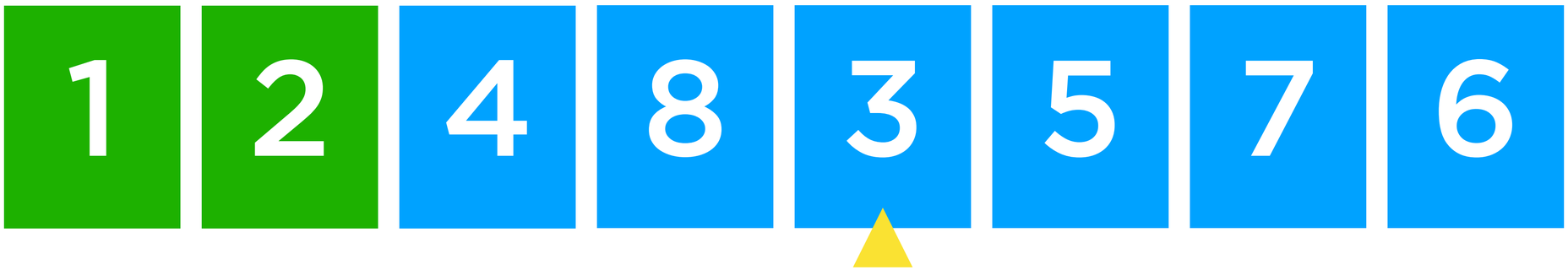

1
3
4
8
2
5
7
6

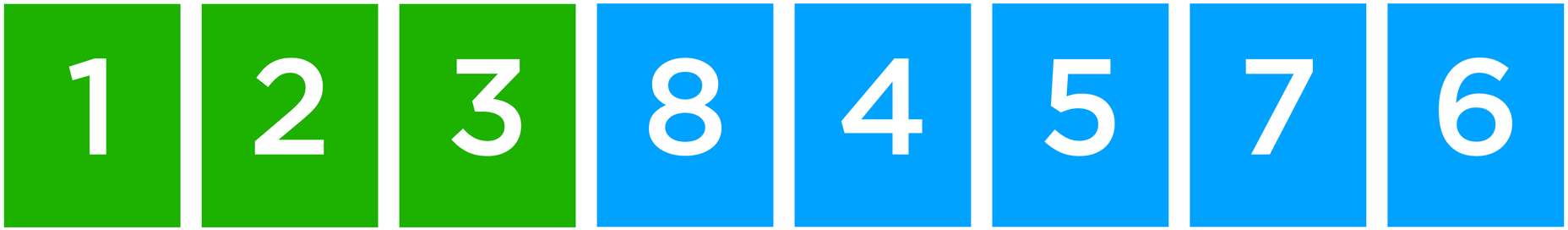

1
3
4
8
2
5
7
6

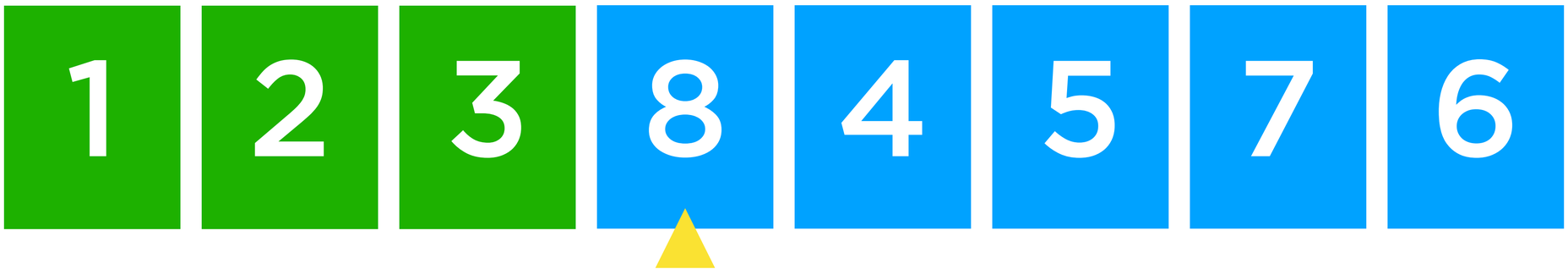

1
3
4
8
2
5
7
6

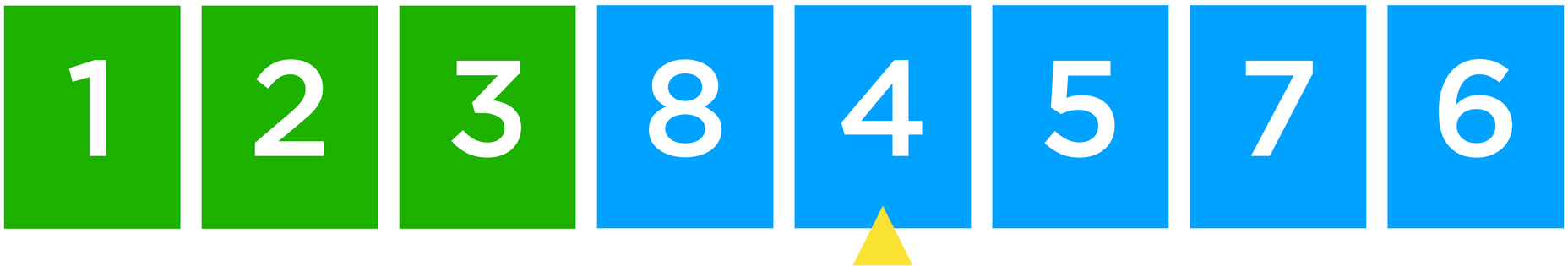

1
2
4
8
3
5
7
6


1
2
4
8
3
5
7
6


1
2
4
8
3
5
7
6


1
2
3
8
4
5
7
6


1
2
3
8
4
5
7
6


1
2
3
8
4
5
7
6


1 2 3 4 8 5 7 6


1
2
3
4
8
5
7
6


1
2
3
4
8
5
7
6


1 2 3 4 5 8 7 6


1
2
3
4
5
8
7
6


1 2 3 4 5 8 7 6


1
2
3
4
5
8
7
6

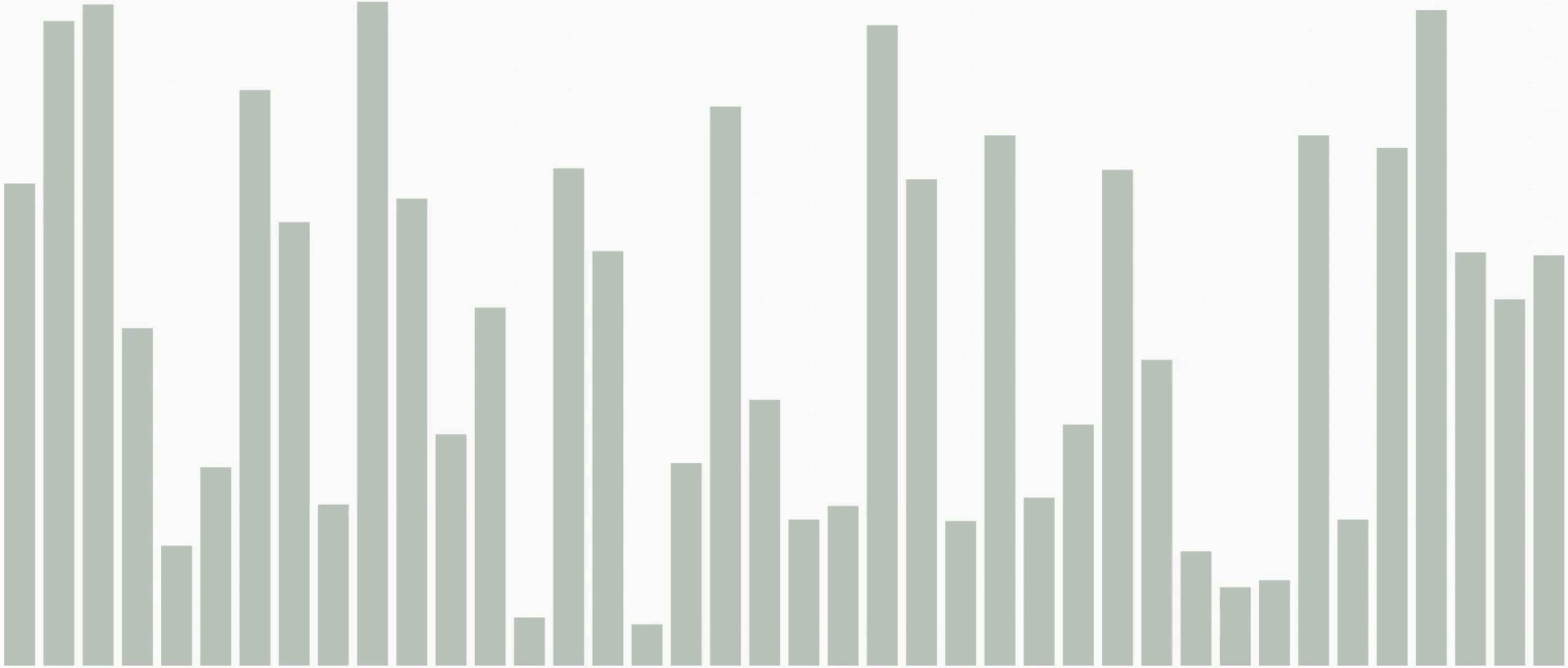

1
2
3
4
5
6
7
8


1
2
3
4
5
6
7
8


1
2
3
4
5
6
7
8


1
2
3
4
5
6
7
8

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
```

Exercise

Complete `sorting.c` such that it sorts integers using Selection Sort.

`cs50.brianyu.me/sorting.c`

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

Part Two

Recursion and Structs

```
void countdown(int n)
{
    if (n == 0)
        return;
    printf("%i\n", n);
    countdown(n - 1);
}
```

```
void countdown(int n)
{
    if (n == 0)
        return;
    printf("%i\n", n);
    countdown(n - 1);
}
```

```
void countdown(int n)
{
    if (n == 0)
        return;
    printf("%i\n", n);
    countdown(n - 1);
}
```

base case

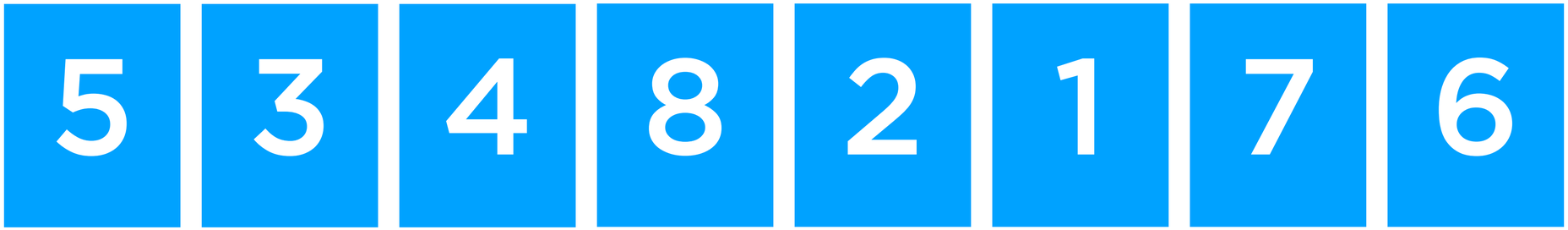
```
void countdown(int n)
{
    if (n == 0)
        return;
    printf("%i\n", n);
    countdown(n - 1);
}
```

```
void countdown(int n)
{
    if (n == 0)
        return;
    printf("%i\n", n);
    countdown(n - 1);
}
```

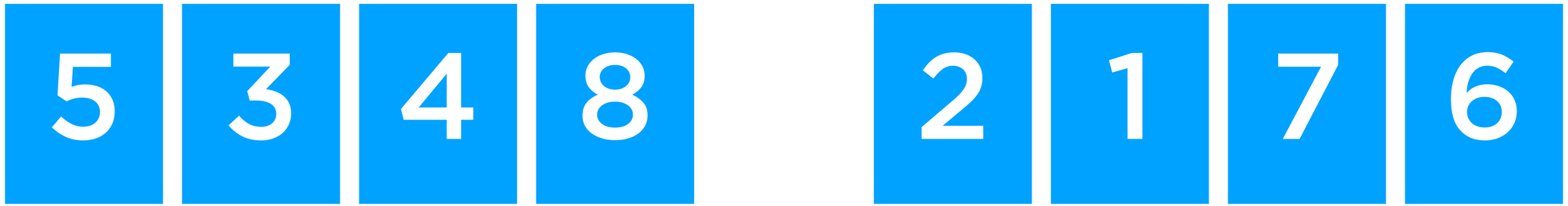
recursive call

```
void countdown(int n)
{
    if (n == 0)
        return;
    printf("%i\n", n);
    countdown(n - 1);
}
```

Mergesort

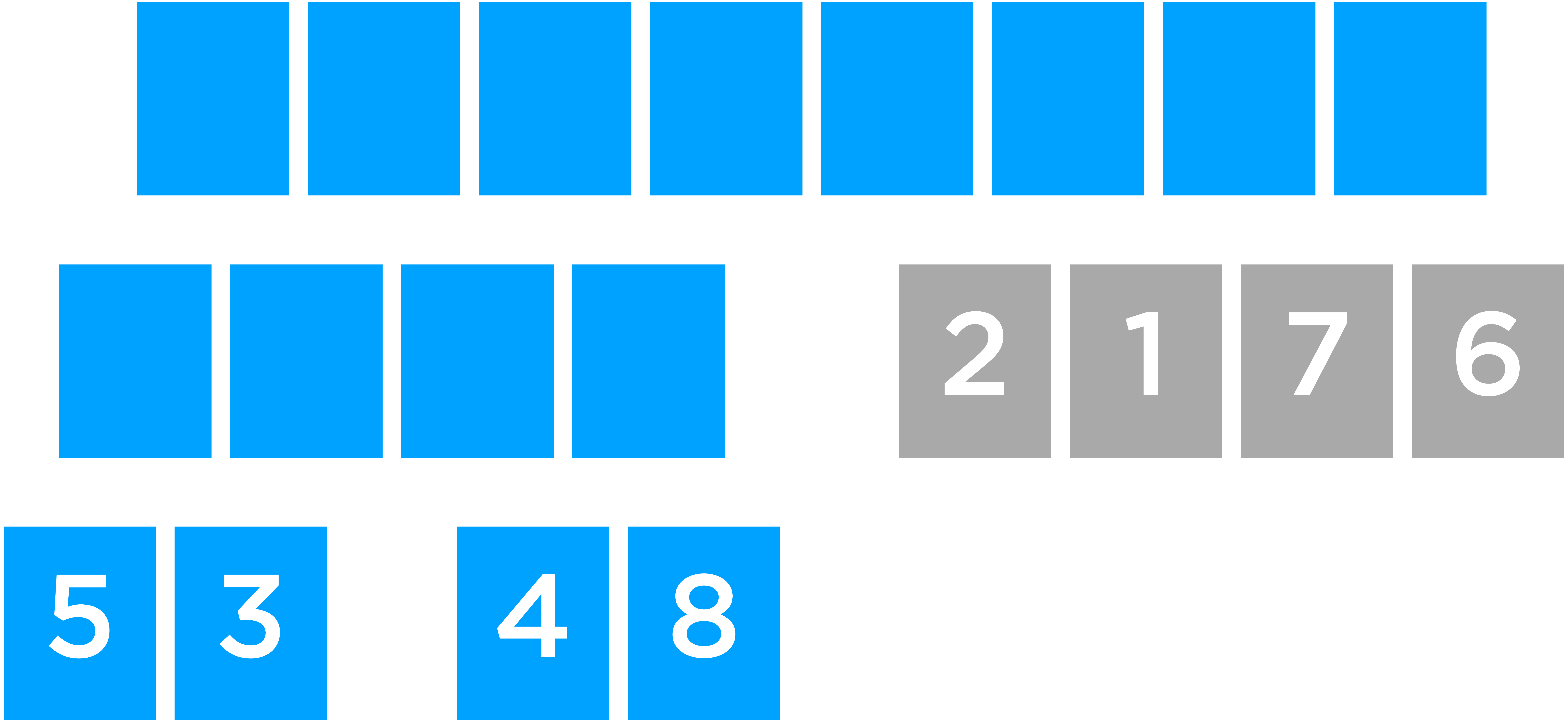

5 3 4 8 2 1 7 6


5 3 4 8 2 1 7 6


5 3 4 8
2 1 7 6

5 3 4 8

2 1 7 6


2
1
7
6
5
3
4
8

2 1 7 6

5 3

4 8

2176

48

53

2176

48

5

3

2176

48

5

3

2176

48

5

3

2 1 7 6

4 8

5 3

2176

48

5 3

2176

3

48

5

2 1 7 6

3 5

4 8

2 1 7 6

3 5

4 8

2 1 7 6

3 5

4 8

2 1 7 6

3 5

4 8

2 1 7 6

3 5

8

2 1 7 6

3 5

4

8

2 1 7 6

3 5

4

8

2 1 7 6

3 5

4 8

2 1 7 6

3 5

4 8


2
1
7
6


3
5


4


8

2 1 7 6

3 5 4 8

2 1 7 6

3 5

4 8

2 1 7 6

3 5

4 8


3


2
1
7
6


5
4
8


3


2
1
7
6


5
4
8


3
4


2
1
7
6


5
8


3
4


2
1
7
6


5
8


3
4
5


2
1
7
6


8

3 4 5 8

2 1 7 6


3
4
5
8


2
1
7
6


3
4
5
8


2
1
7
6


3
4
5
8


2
1


7
6

3 4 5 8

2 1

7 6

3 4 5 8

7 6

2 1

3 4 5 8

7 6

2

1

3 4 5 8

7 6

2

1

3 4 5 8

7 6

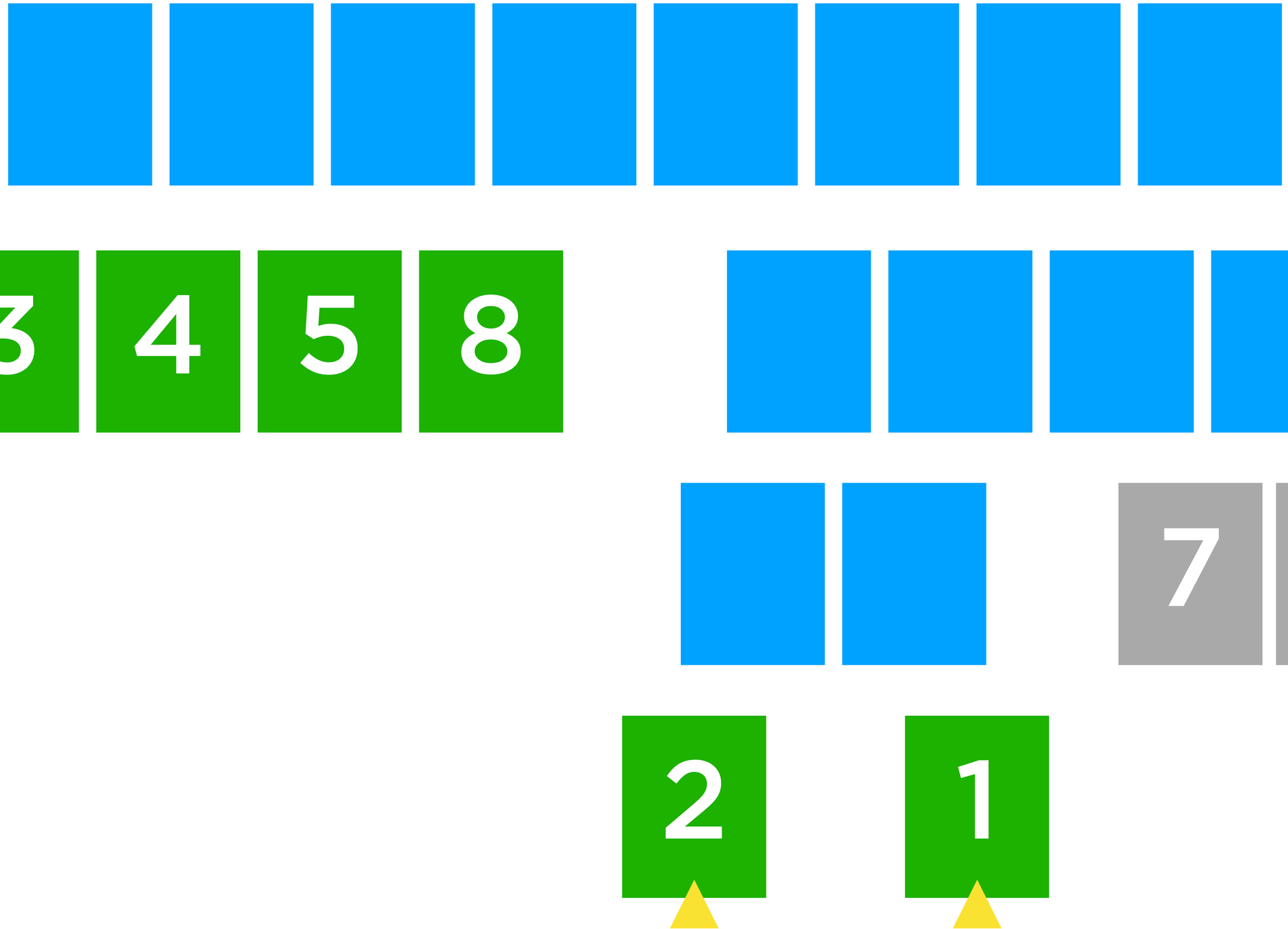
2

1

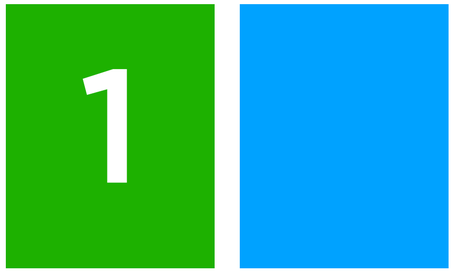
3 4 5 8

7 6

2 1


3
4
5
8
7
6
2
1


3
4
5
8


1


7
6


2


3
4
5
8


1
2


7
6

3 4 5 8

1 2

7 6


3
4
5
8


1
2


7
6


3
4
5
8


1
2


7


6


3
4
5
8


1
2


7


6


3
4
5
8


1
2


7


6


3
4
5
8


1
2


7


6

3 4 5 8

1 2

7

6


3
4
5
8


1
2


7


6


3
4
5
8


1
2


6

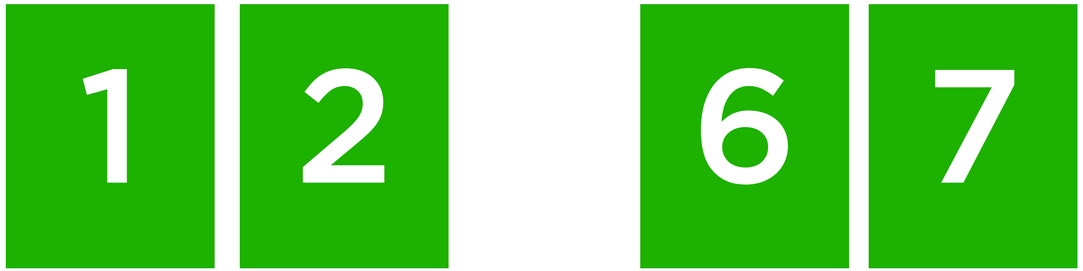

7

3 4 5 8

1 2

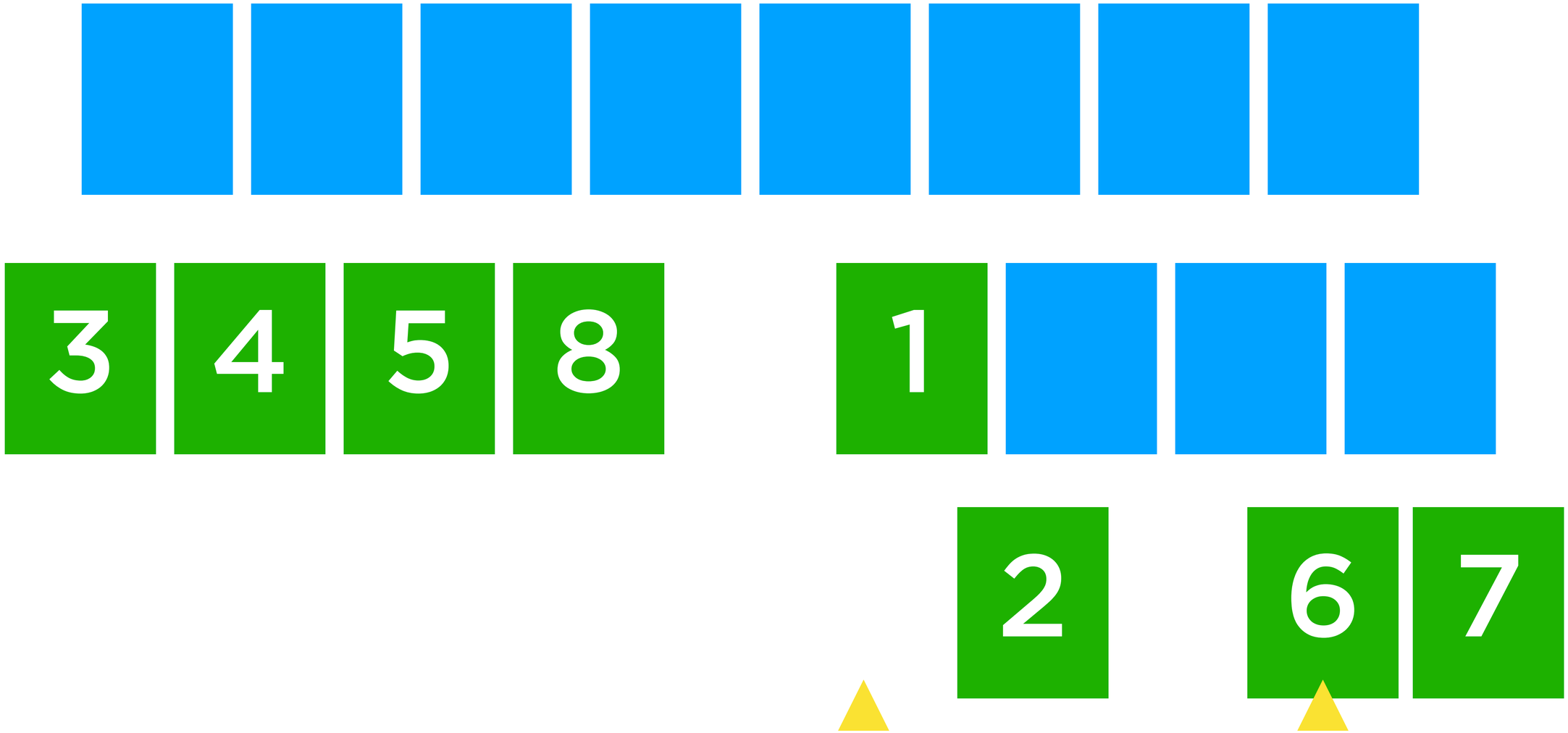
6 7


3
4
5
8

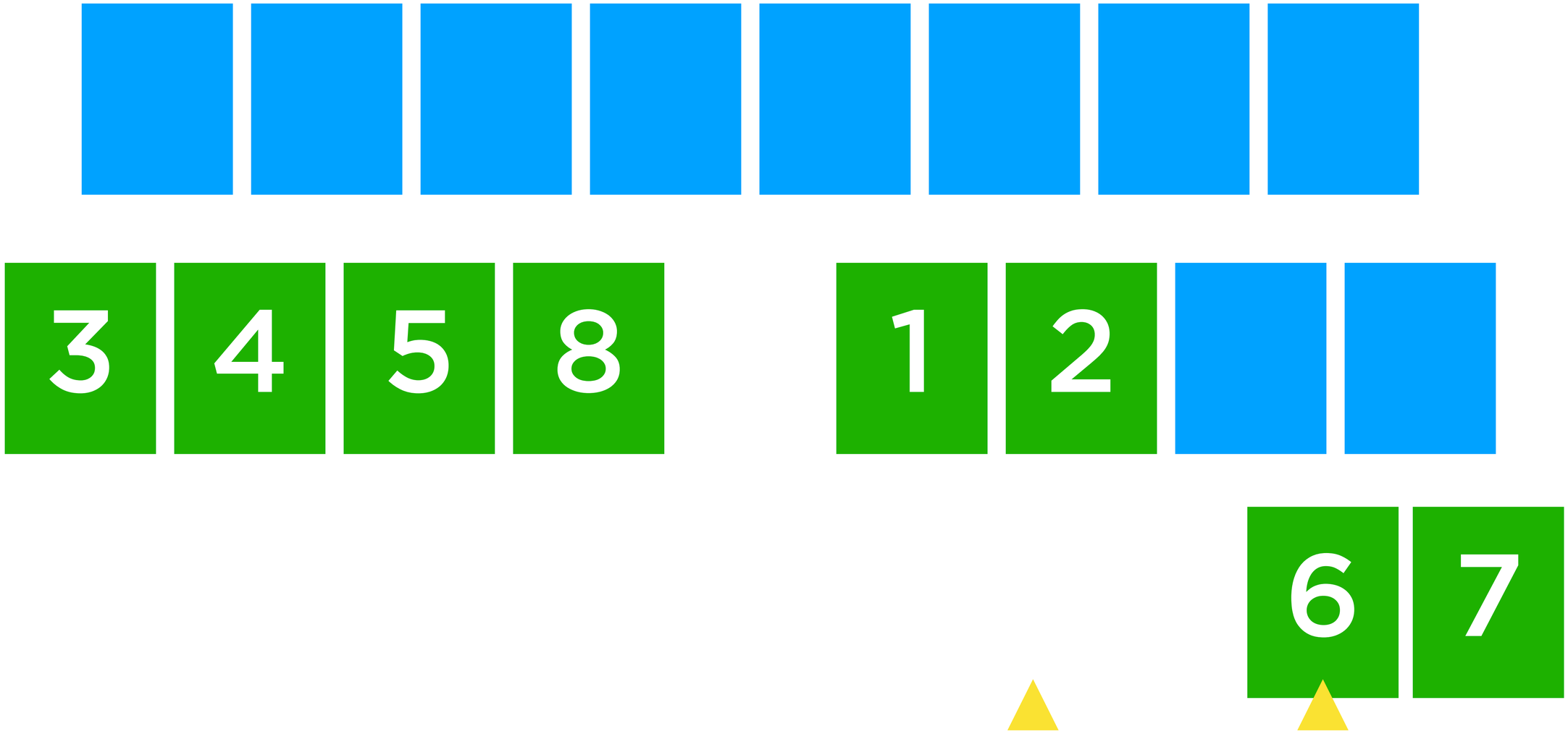

1
2
6
7


3
4
5
8


1
2
6
7


3
4
5
8
1
2
6
7

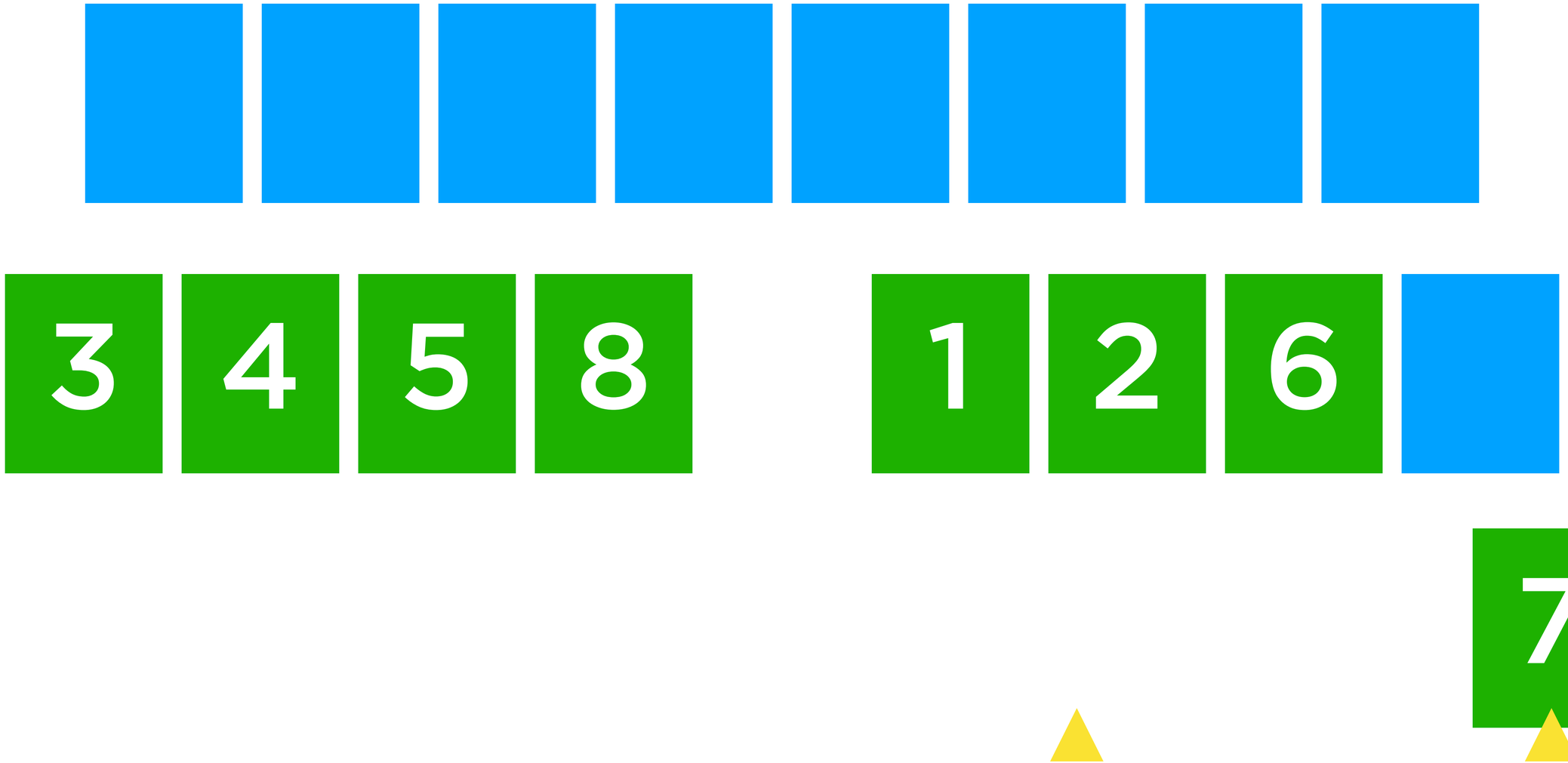

3
4
5
8
1
2
6
7


3
4
5
8
1
2
6
7


3
4
5
8


1
2
6


7

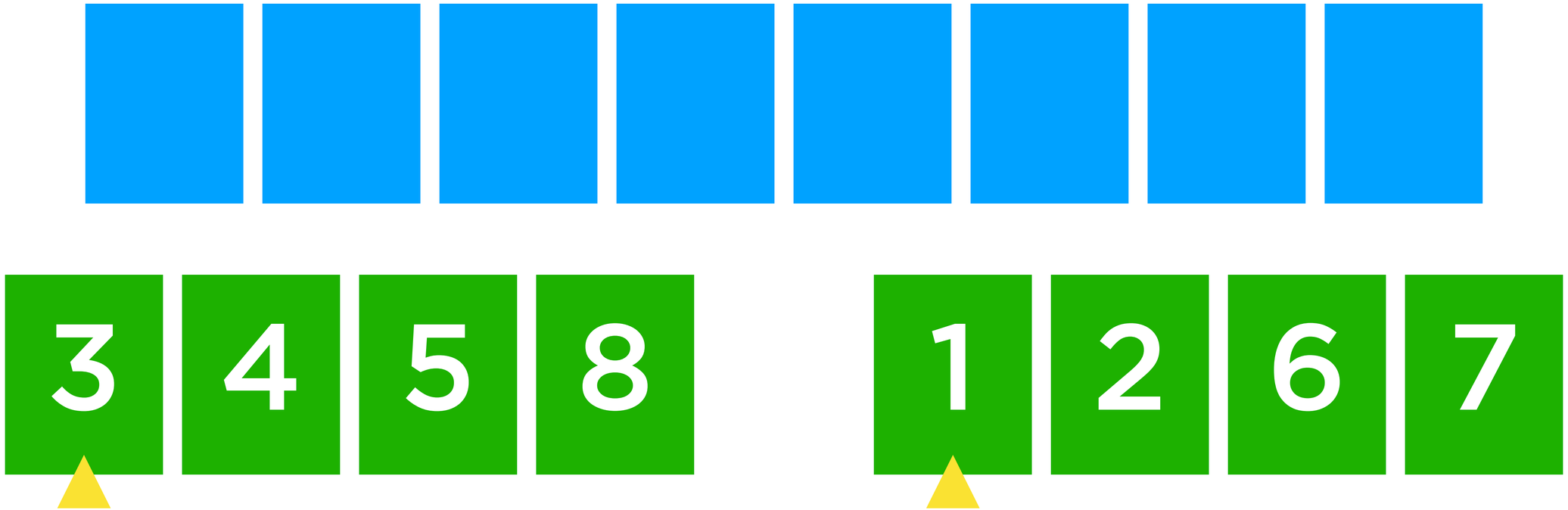

3 4 5 8
1 2 6
7

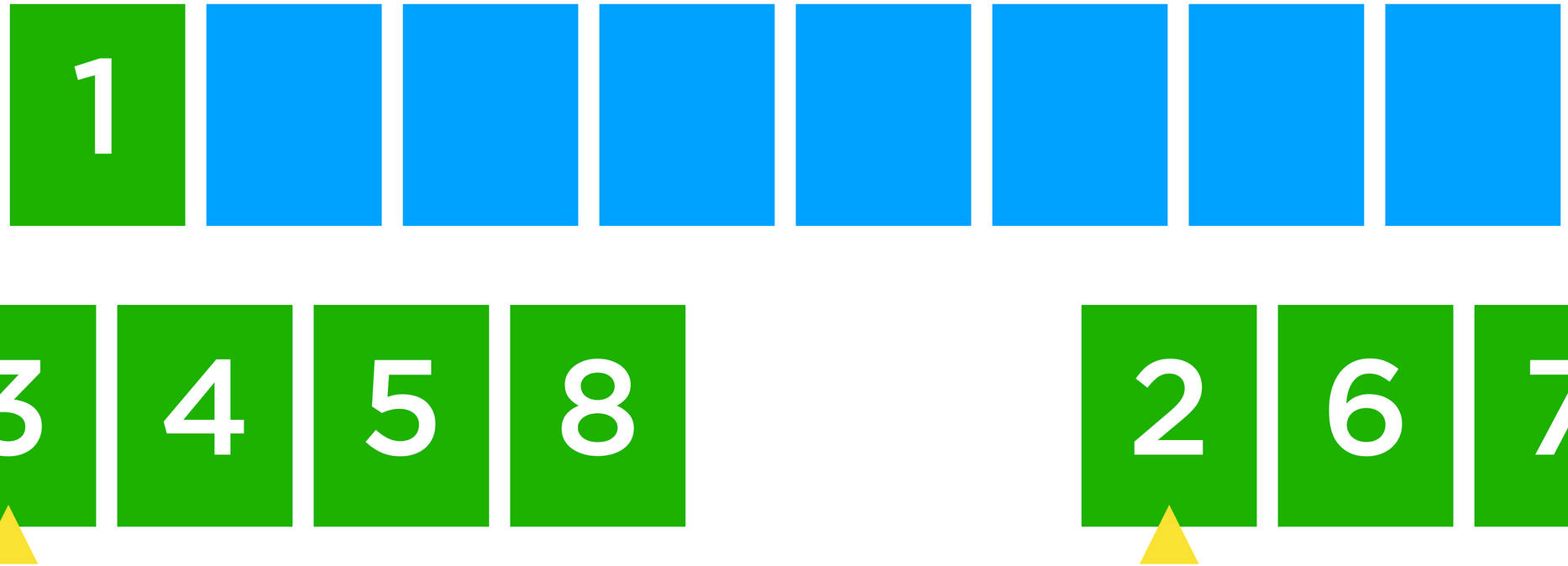
3 4 5 8

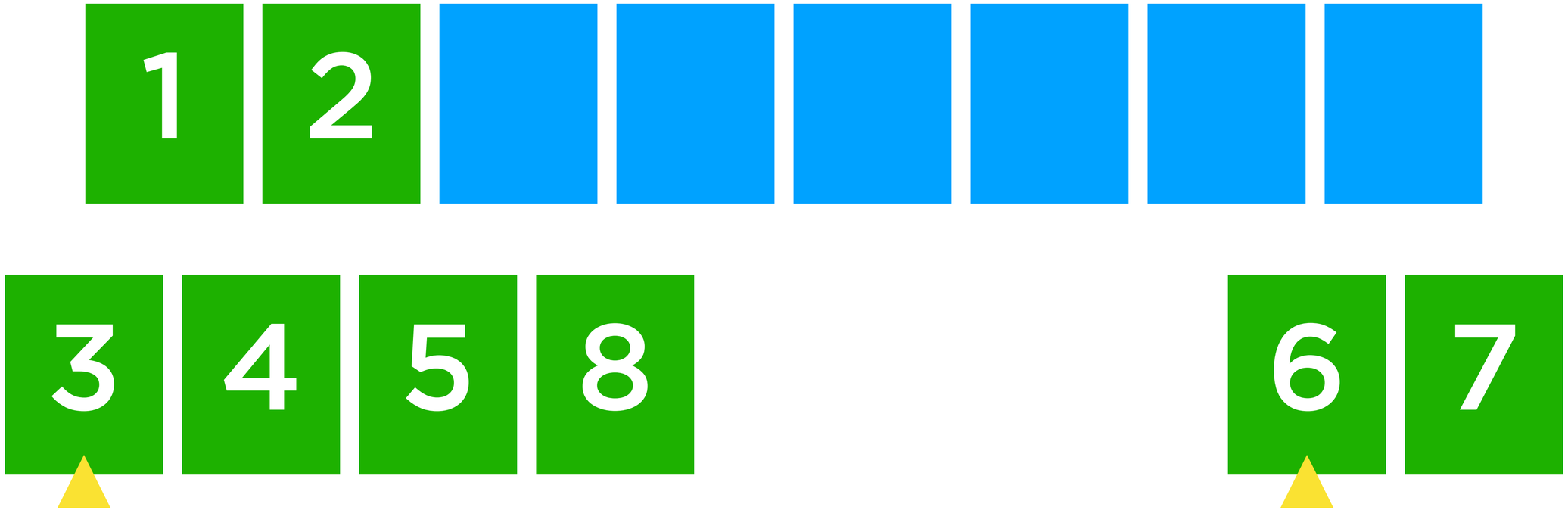
1 2 6 7

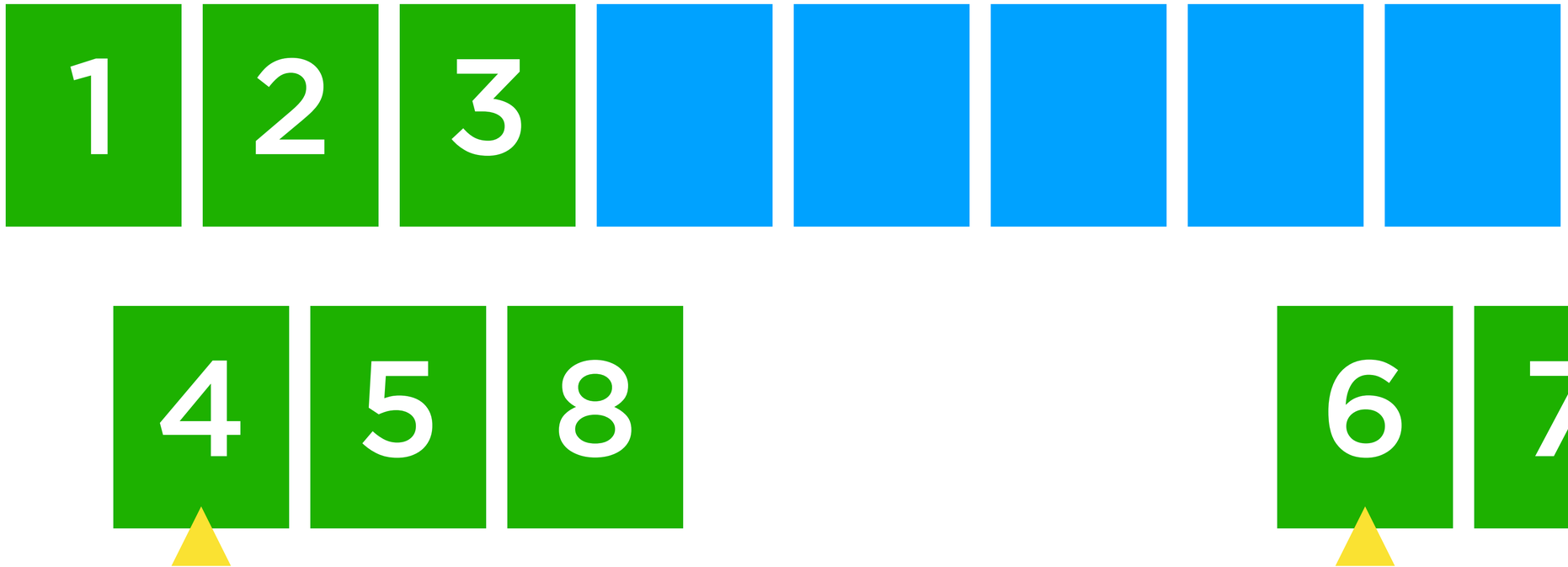
3 4 5 8

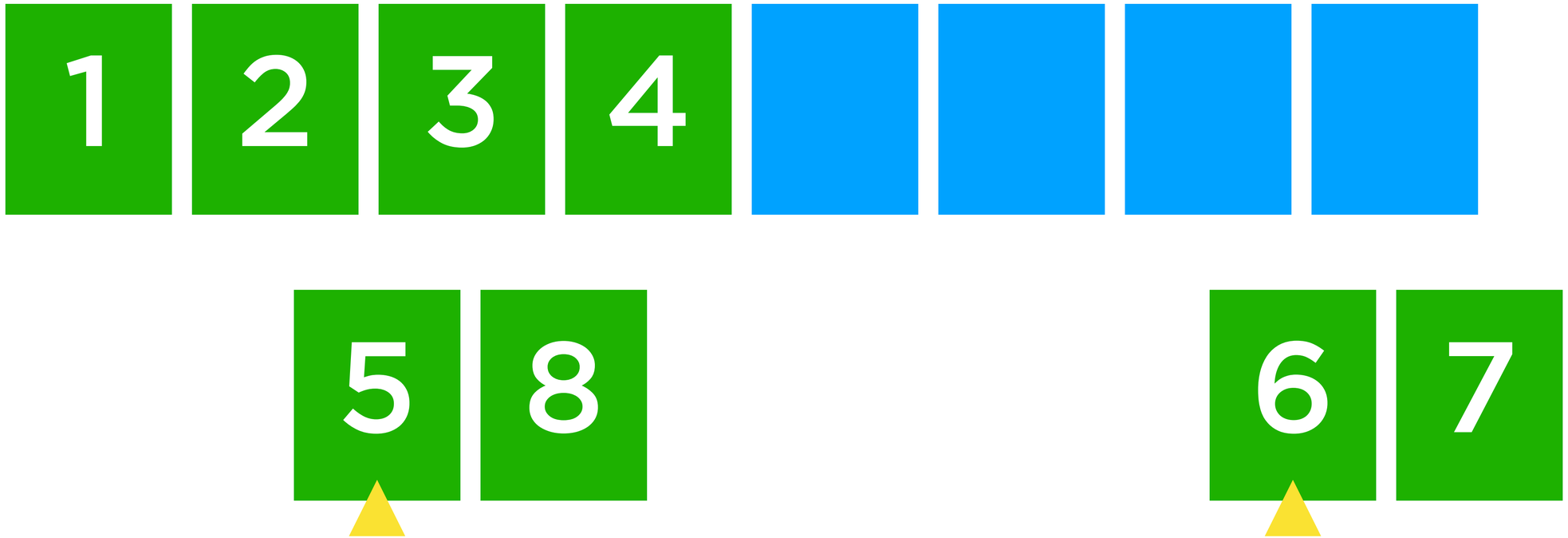
1 2 6 7

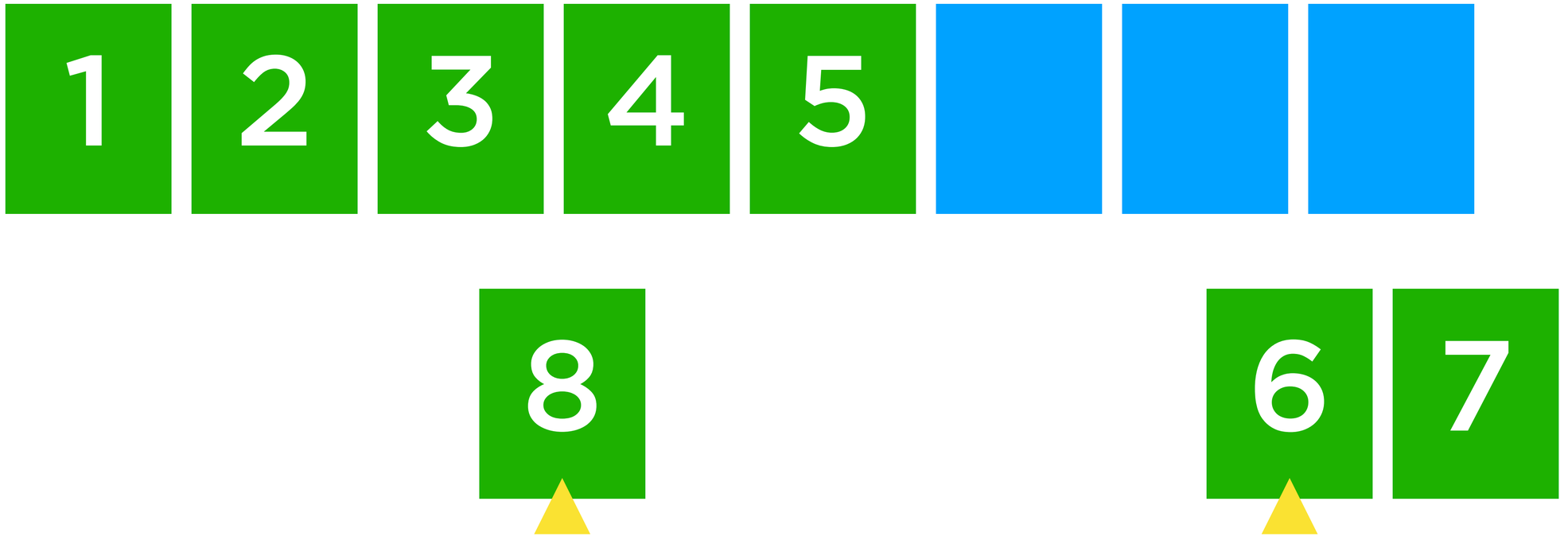

3 4 5 8
1 2 6 7

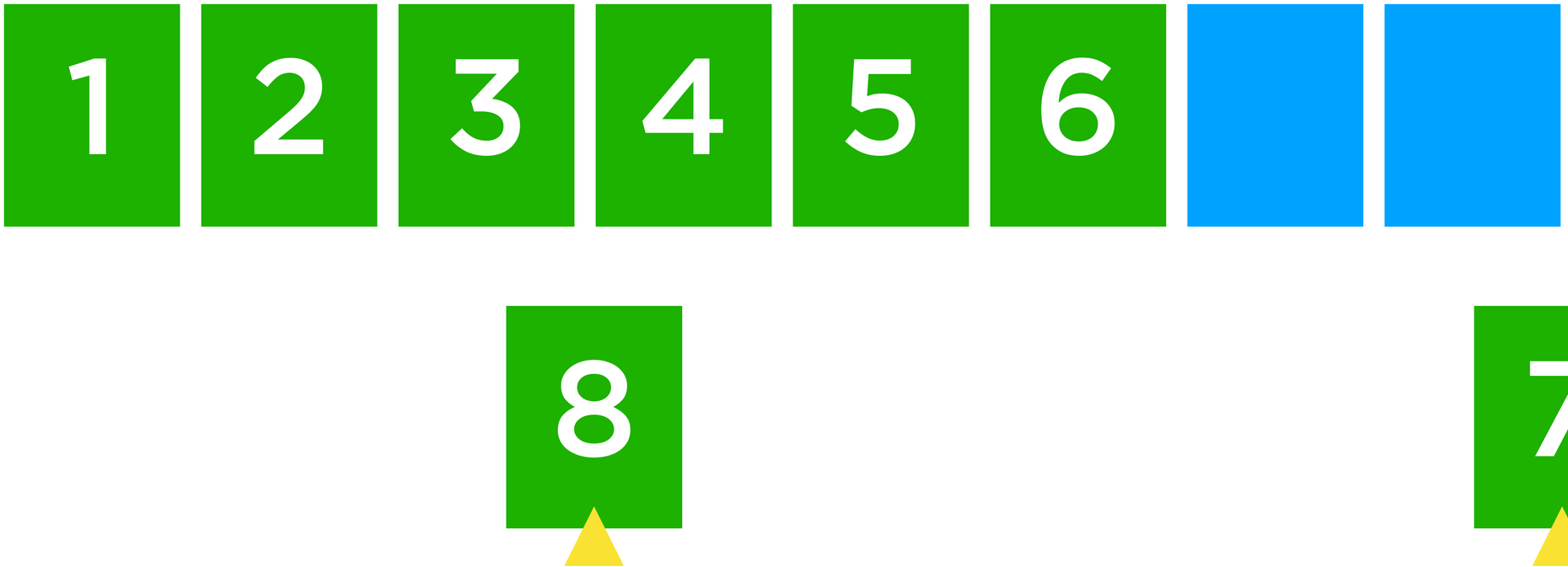

1
3
4
5
8
2
6
7

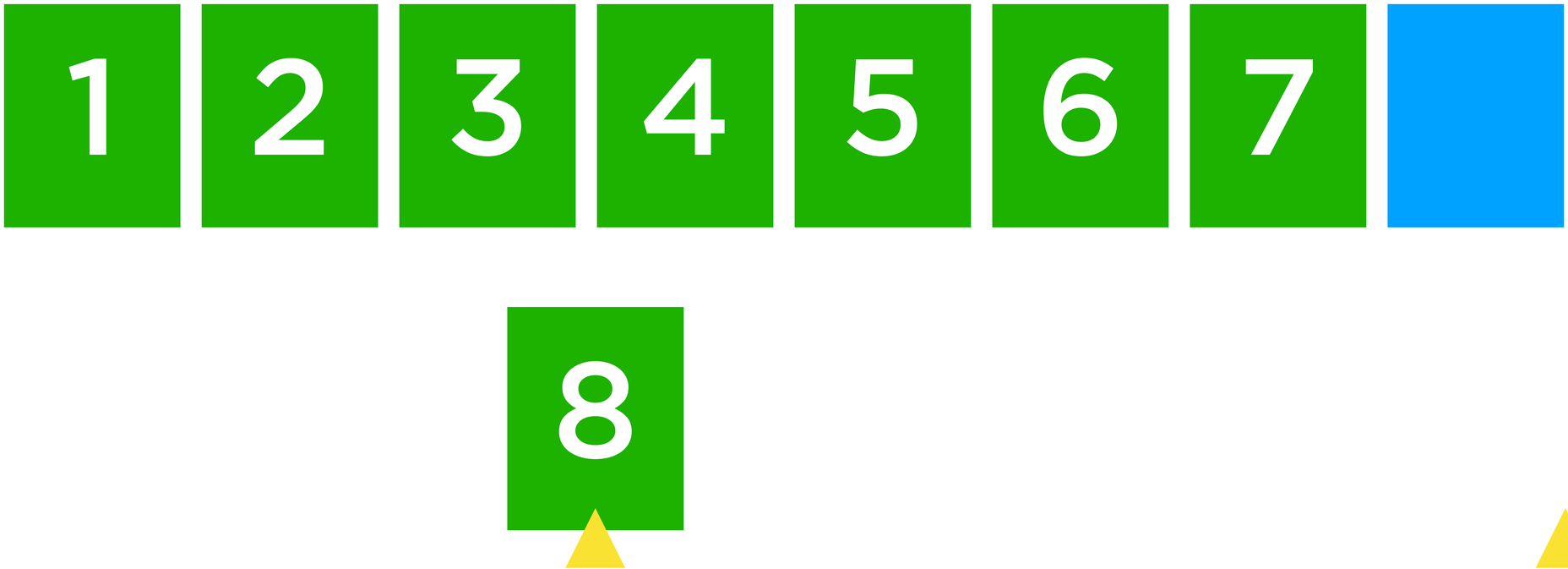

1
2
3
4
5
8
6
7


1
2
3
4
5
8
6
7


1
2
3
4
5
8
6
7


1
2
3
4
5
8
6
7


1
2
3
4
5
6
8
7

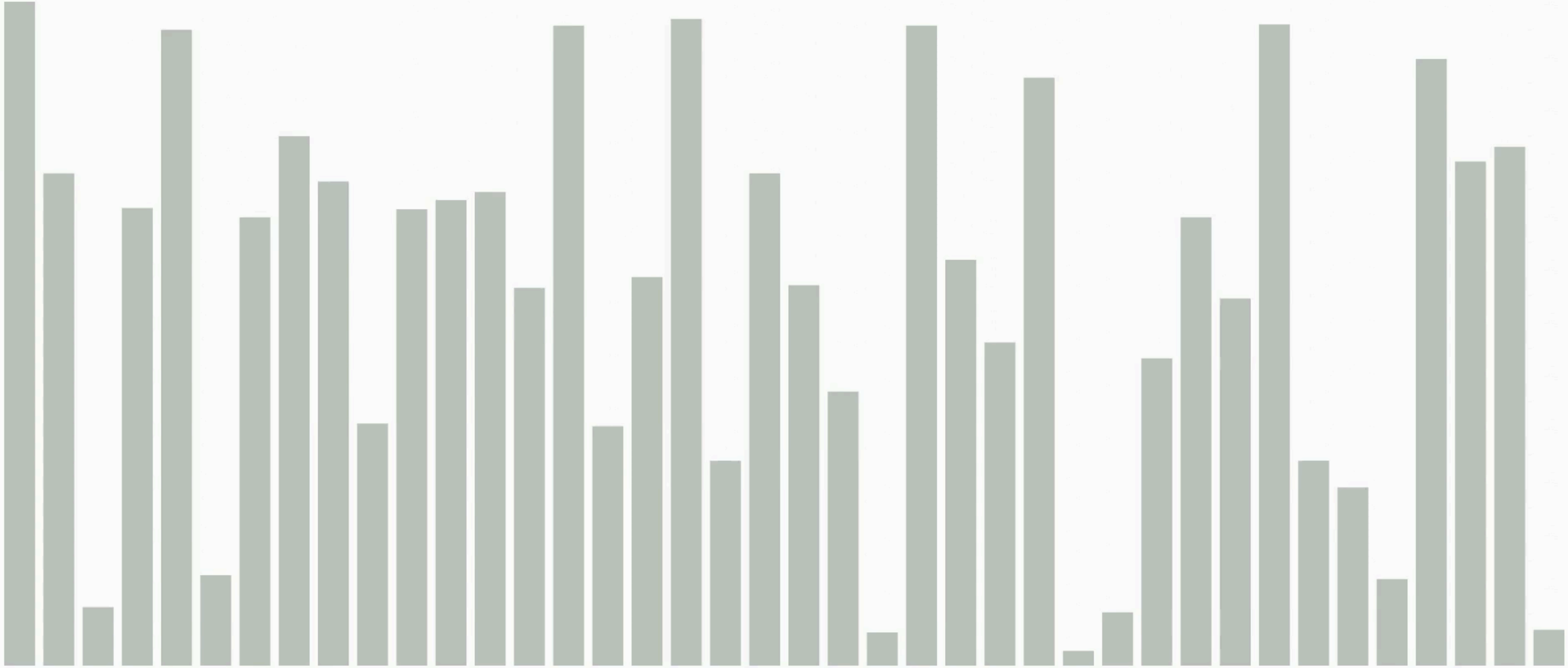

1
2
3
4
5
6
7
8


1
2
3
4
5
6
7
8

1 2 3 4 5 6 7 8


1
2
3
4
5
6
7
8

Mergesort



Structs

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

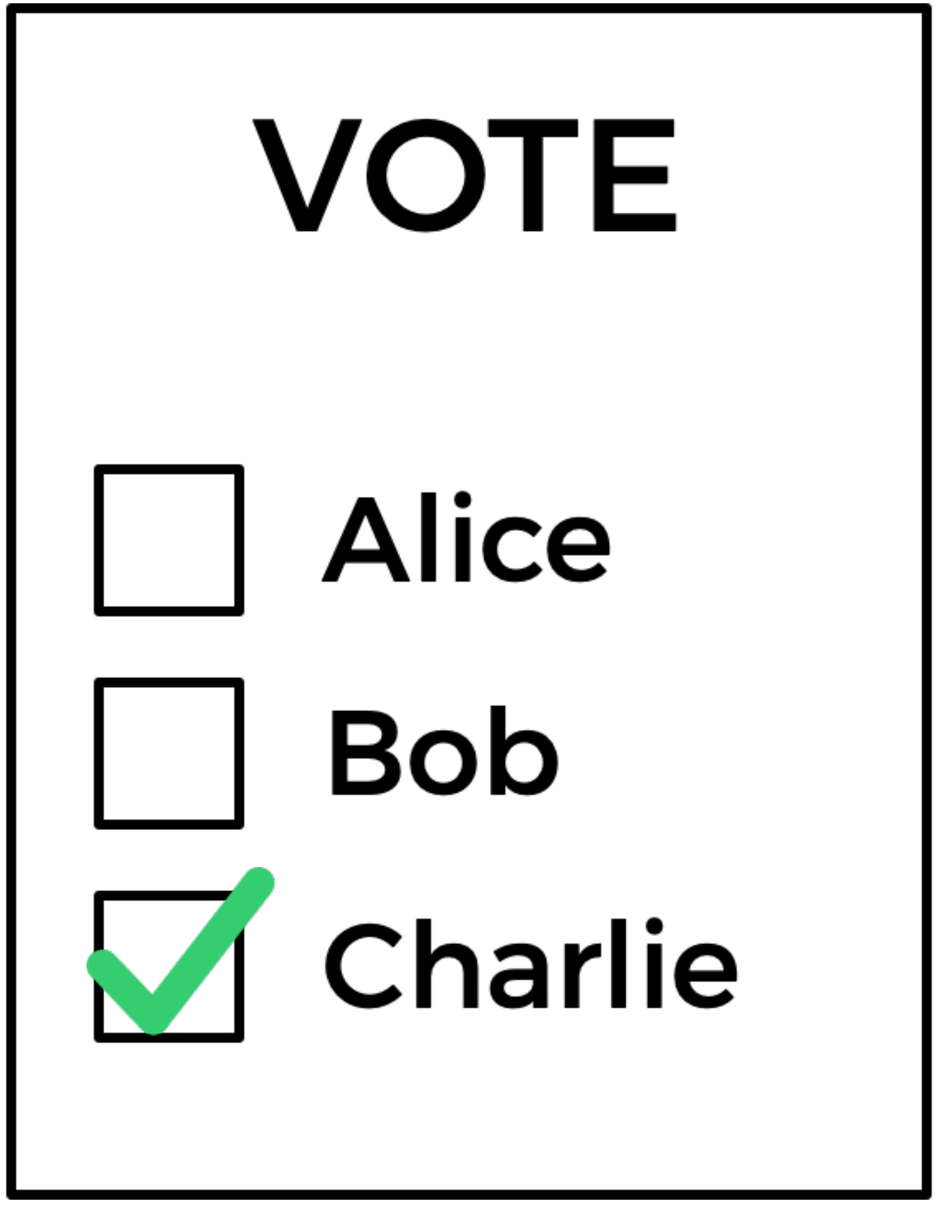
```
typedef struct
{
    string name;
    string number;
}
person;
```

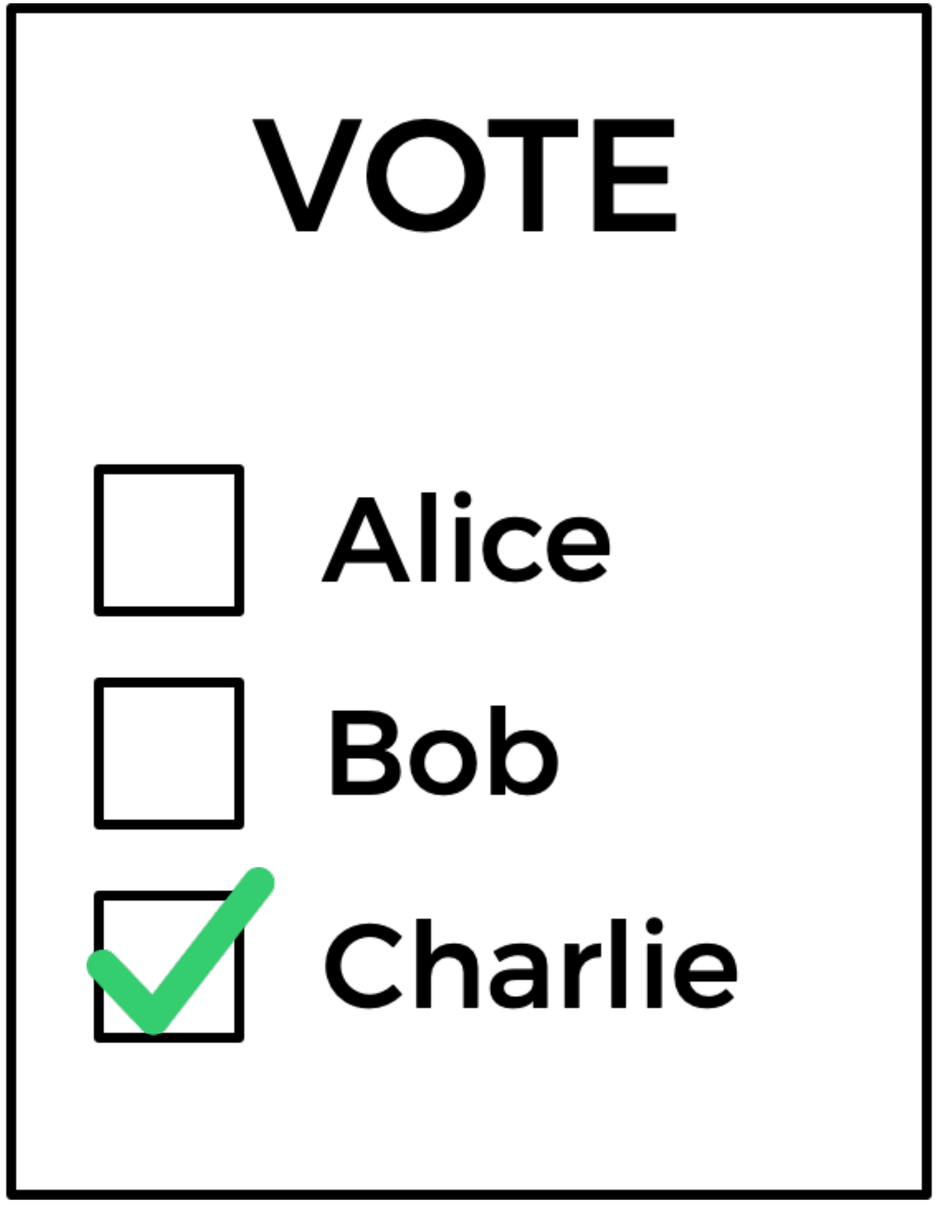
```
person p;

p.name = "Emma";

p.number = "555-0100";
```

Problem Set 3

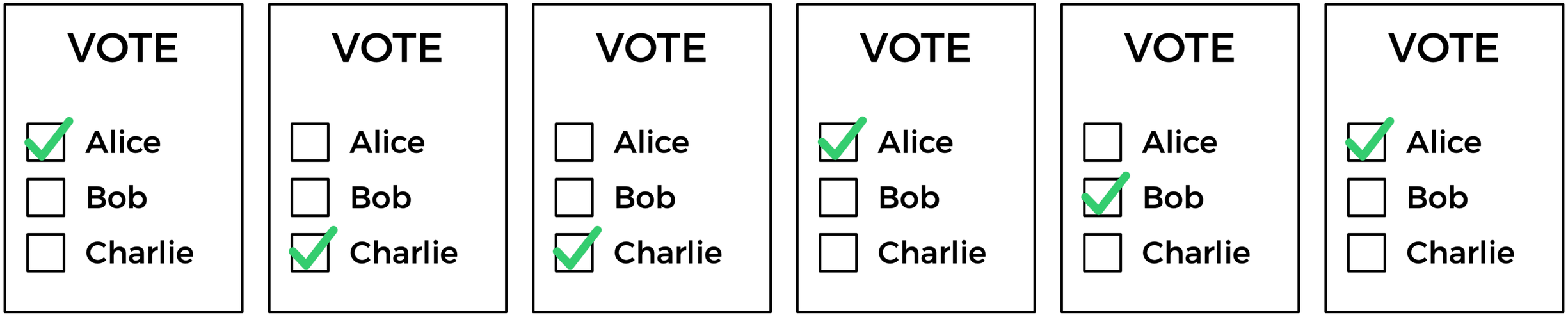
Problem Set 3

- Plurality
- One of:
    - Runoff
    - Tideman

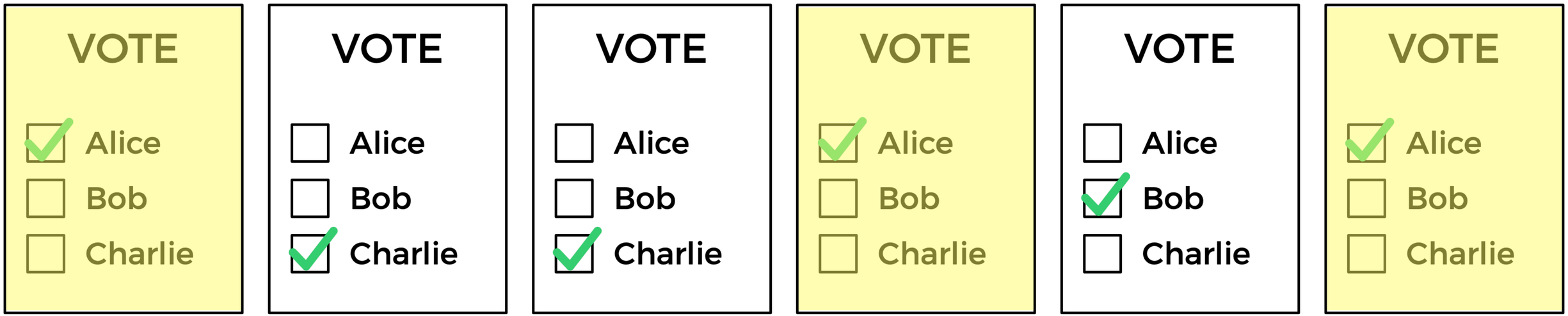

VOTE
Alice
Bob
Charlie


VOTE
Alice
Bob
Charlie

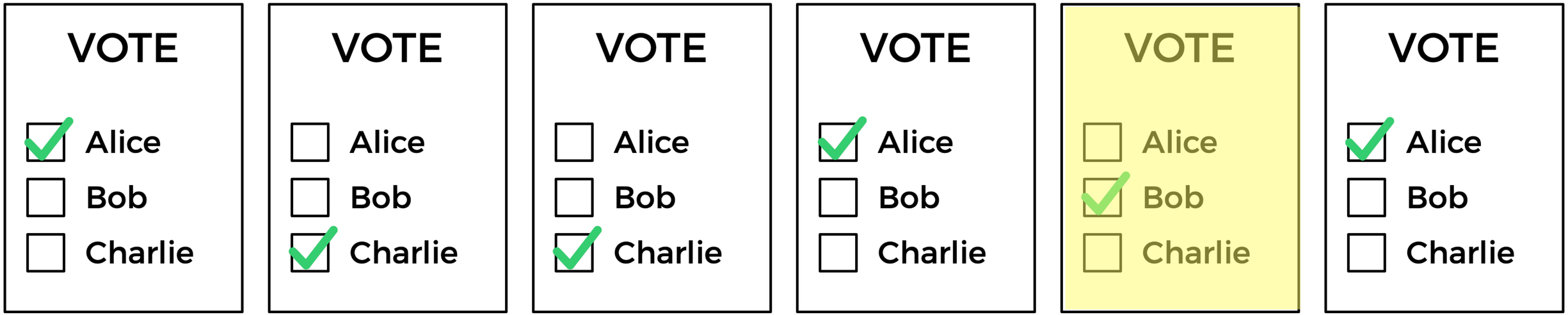
name	Alice	Bob	Charlie
votes			



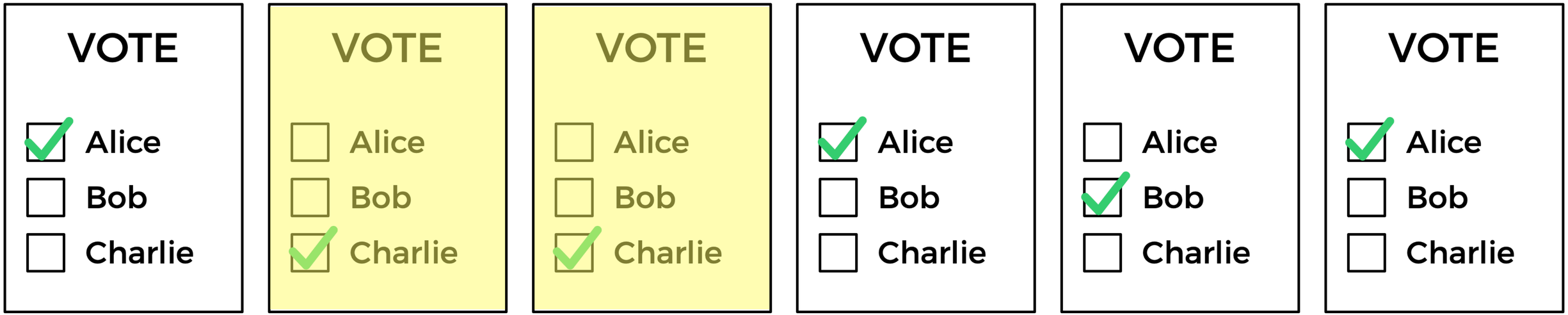
name	Alice	Bob	Charlie
votes	3		



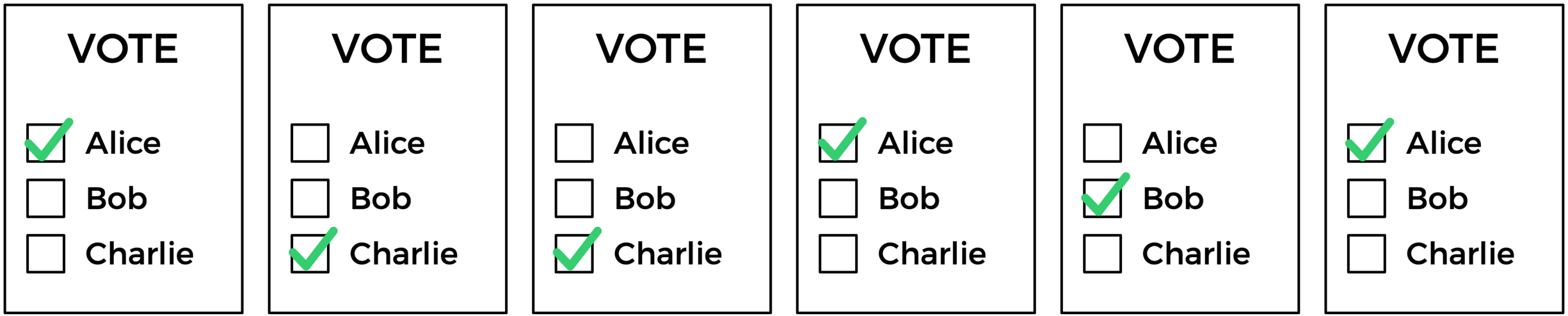
name	Alice	Bob	Charlie
votes	3	1	



name	Alice	Bob	Charlie
votes	3	1	2



name	Alice	Bob	Charlie
votes	3	1	2



```
typedef struct
{
    string name;
    int votes;
}
candidate;

candidate candidates[MAX];
```

name	Alice
votes	3

name	Alice	Bob	Charlie
votes	3	1	2

candidates[0]

name	Alice	Bob	Charlie
votes	3	1	2

candidates[1]

name	Alice	Bob	Charlie
votes	3	1	2

candidates[2]

name	Alice	Bob	Charlie
votes	3	1	2

```
candidates[2].name
```

name	Alice	Bob	Charlie
votes	3	1	2

```
candidates[2].votes
```

Part Three

Lab

This is CS50.