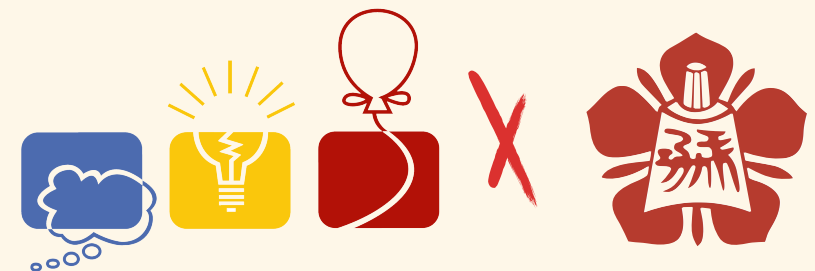


Quick Sort

郭至軒 (KuoE0)

KuoE0.tw@gmail.com

KuoE0.ch





Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)

<http://creativecommons.org/licenses/by-sa/3.0/>

Latest update: Mar 15, 2013

Algorithm

採用 divide & conquer 策略

Divide

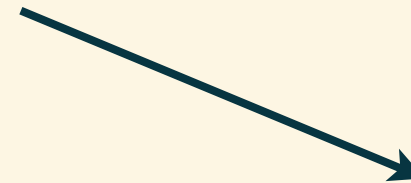
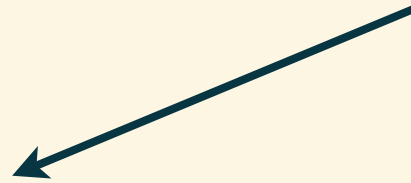
從數列中挑出一個元素作為 **pivot**, 利用 pivot 將原數列分為兩個子數列：

- 所有元素小於 pivot 的數列 A
- 所有元素大於 pivot 的元素的數列 B
- 等於 pivot 的元素可放置在任一個中

2	9	4	3	8	7	5	1	6
---	---	---	---	---	---	---	---	---

2	9	4	3	8	7	5	1	6
---	---	---	---	---	---	---	---	---

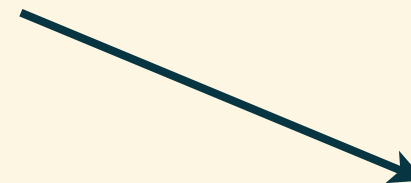
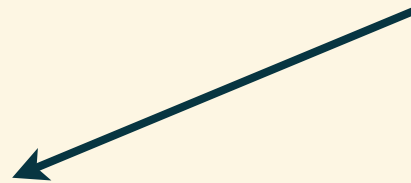
2	9	4	3	8	7	5	1	6
---	---	---	---	---	---	---	---	---



2	4	3	5	1	6
---	---	---	---	---	---

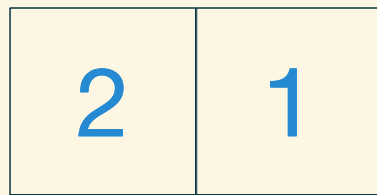
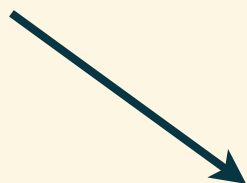
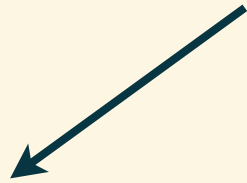
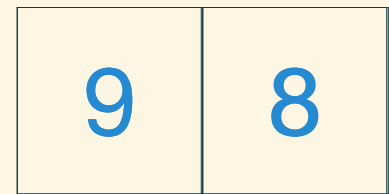
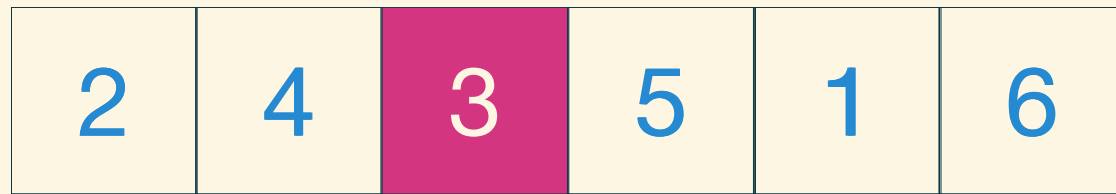
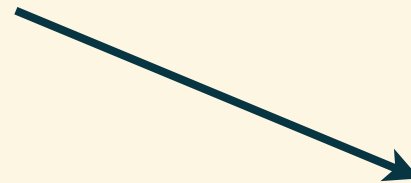
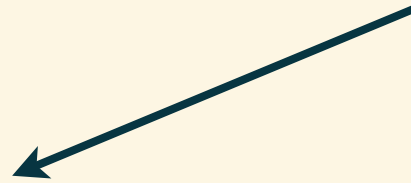
9	8
---	---

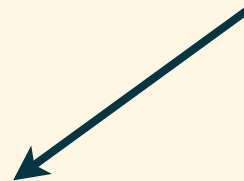
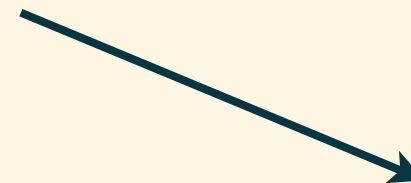
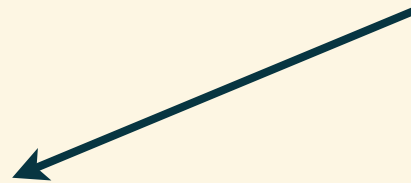
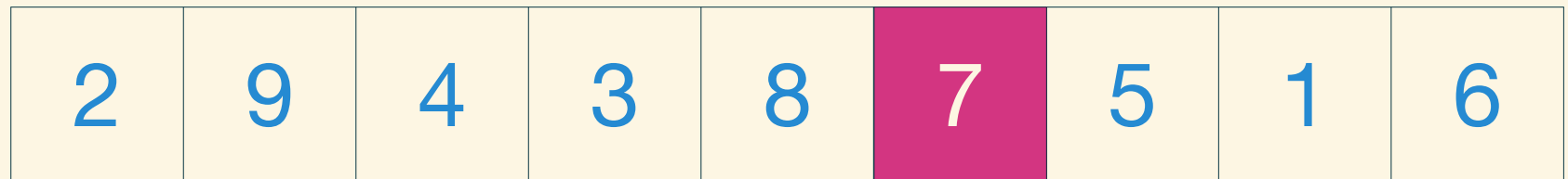
2	9	4	3	8	7	5	1	6
---	---	---	---	---	---	---	---	---

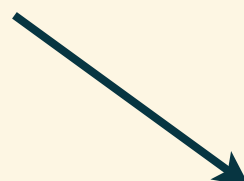
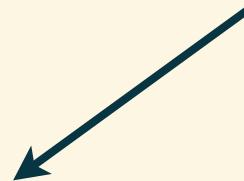
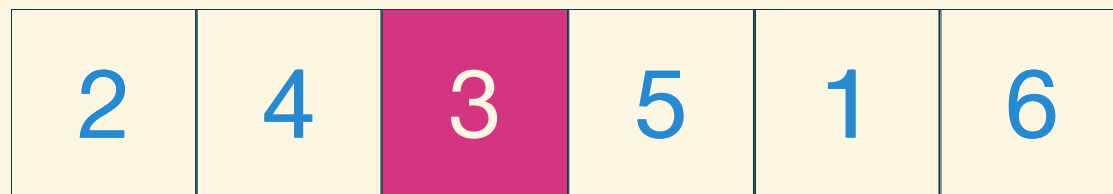
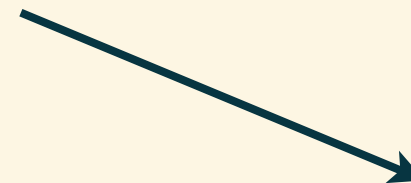
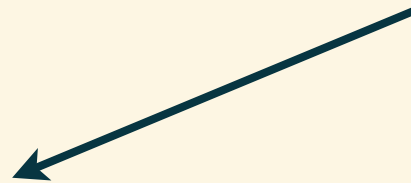
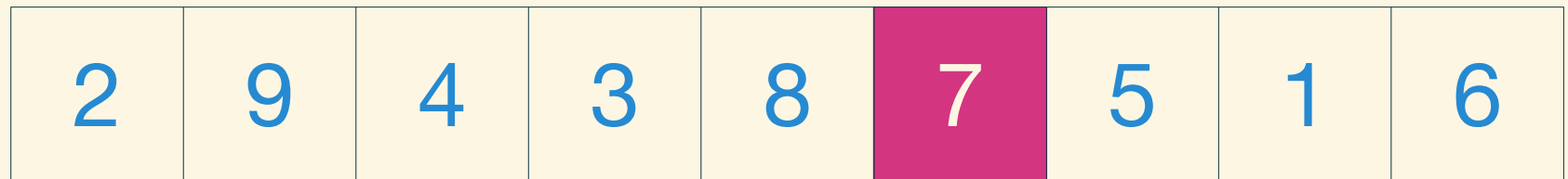


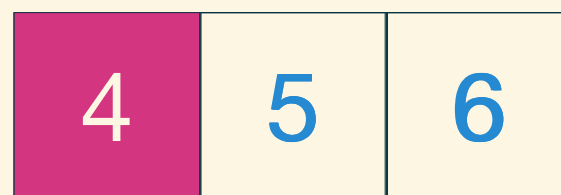
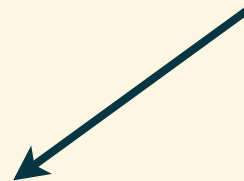
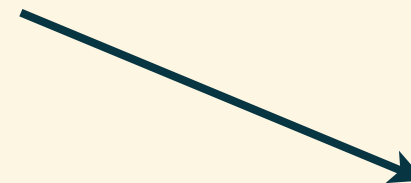
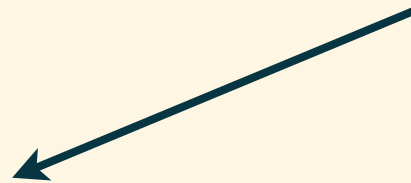
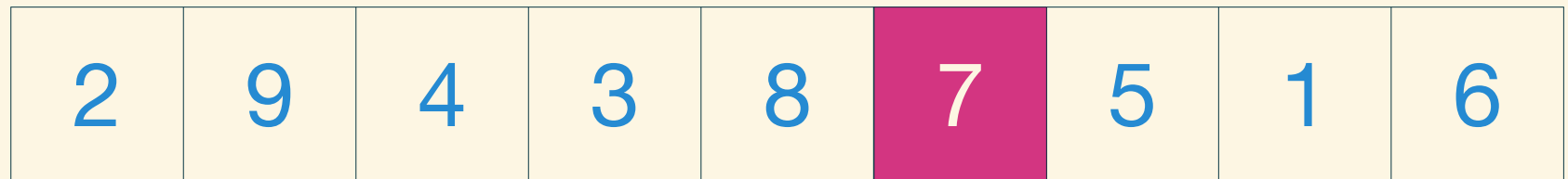
2	4	3	5	1	6
---	---	---	---	---	---

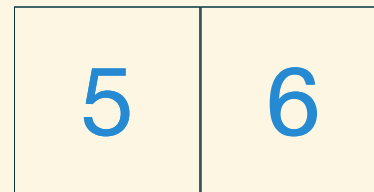
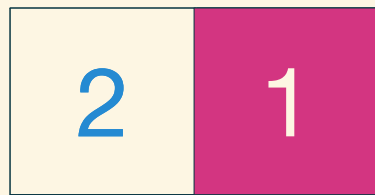
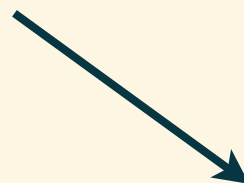
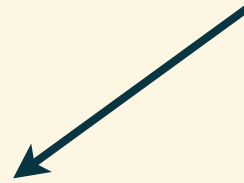
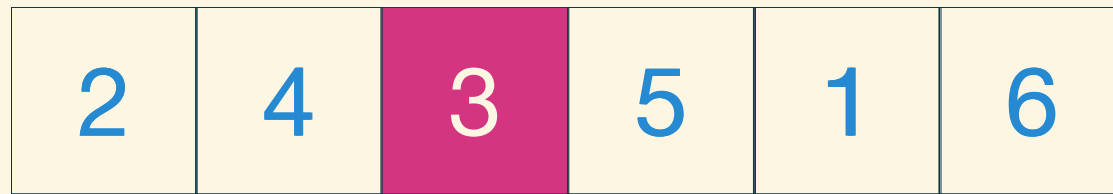
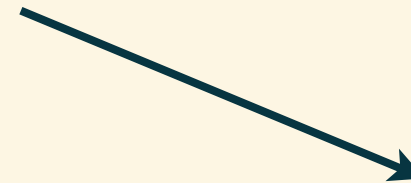
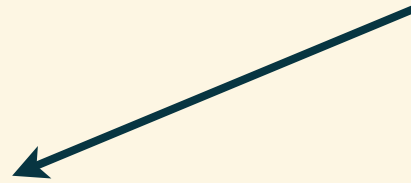
9	8
---	---

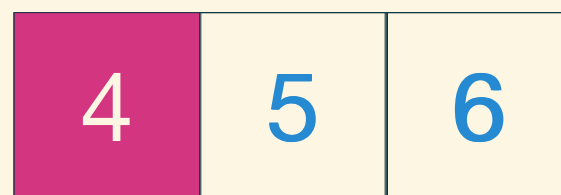
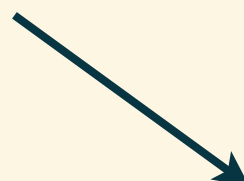
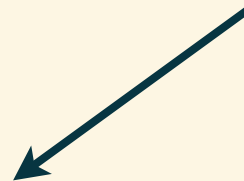
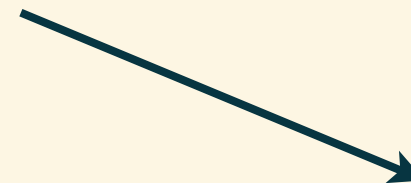
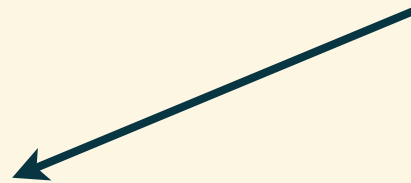
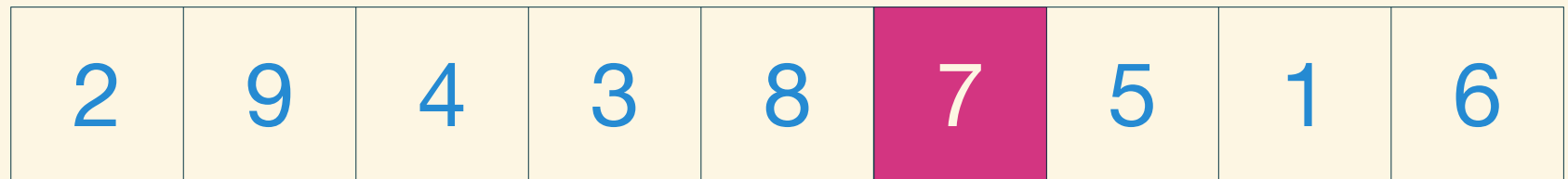


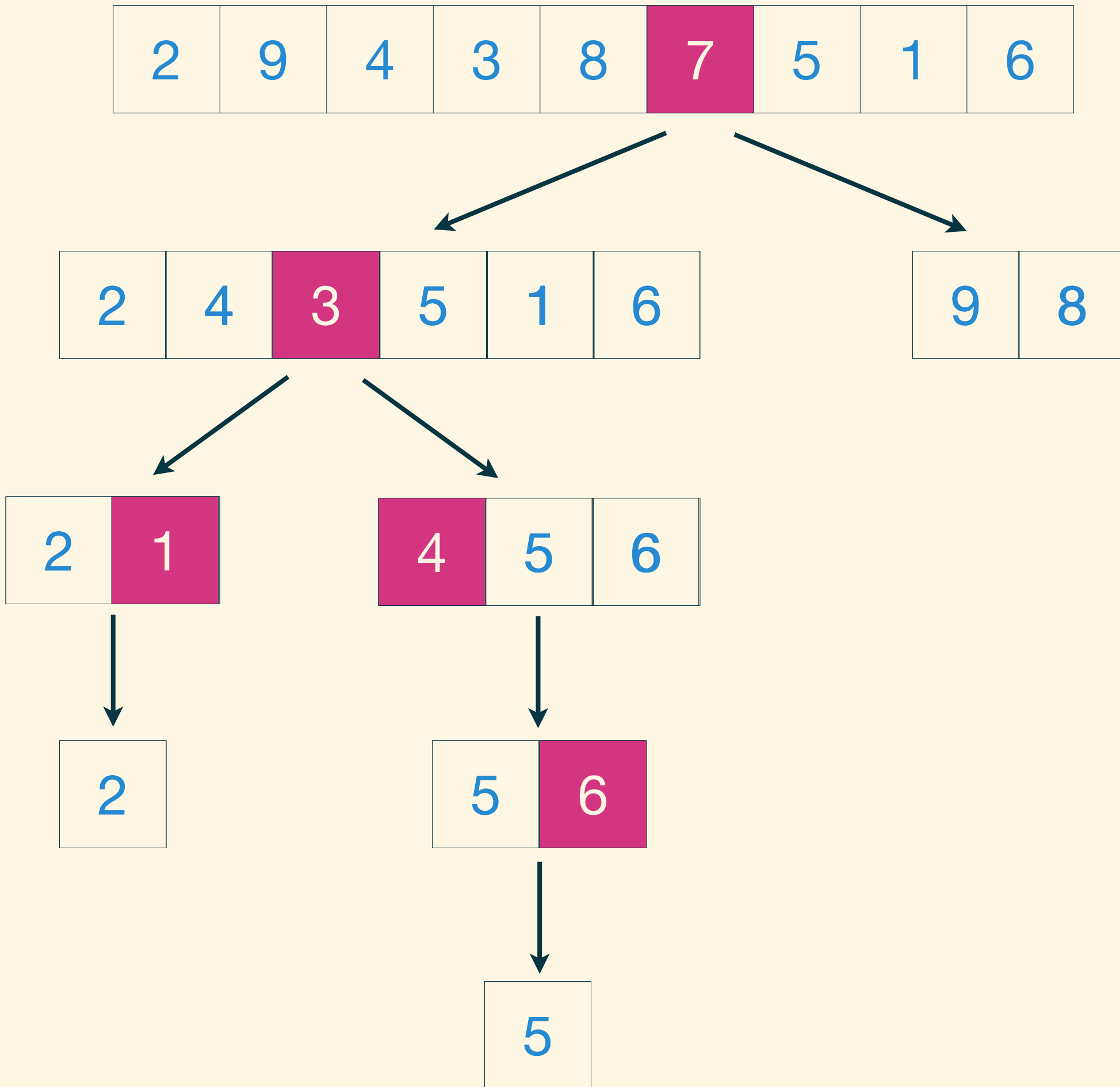


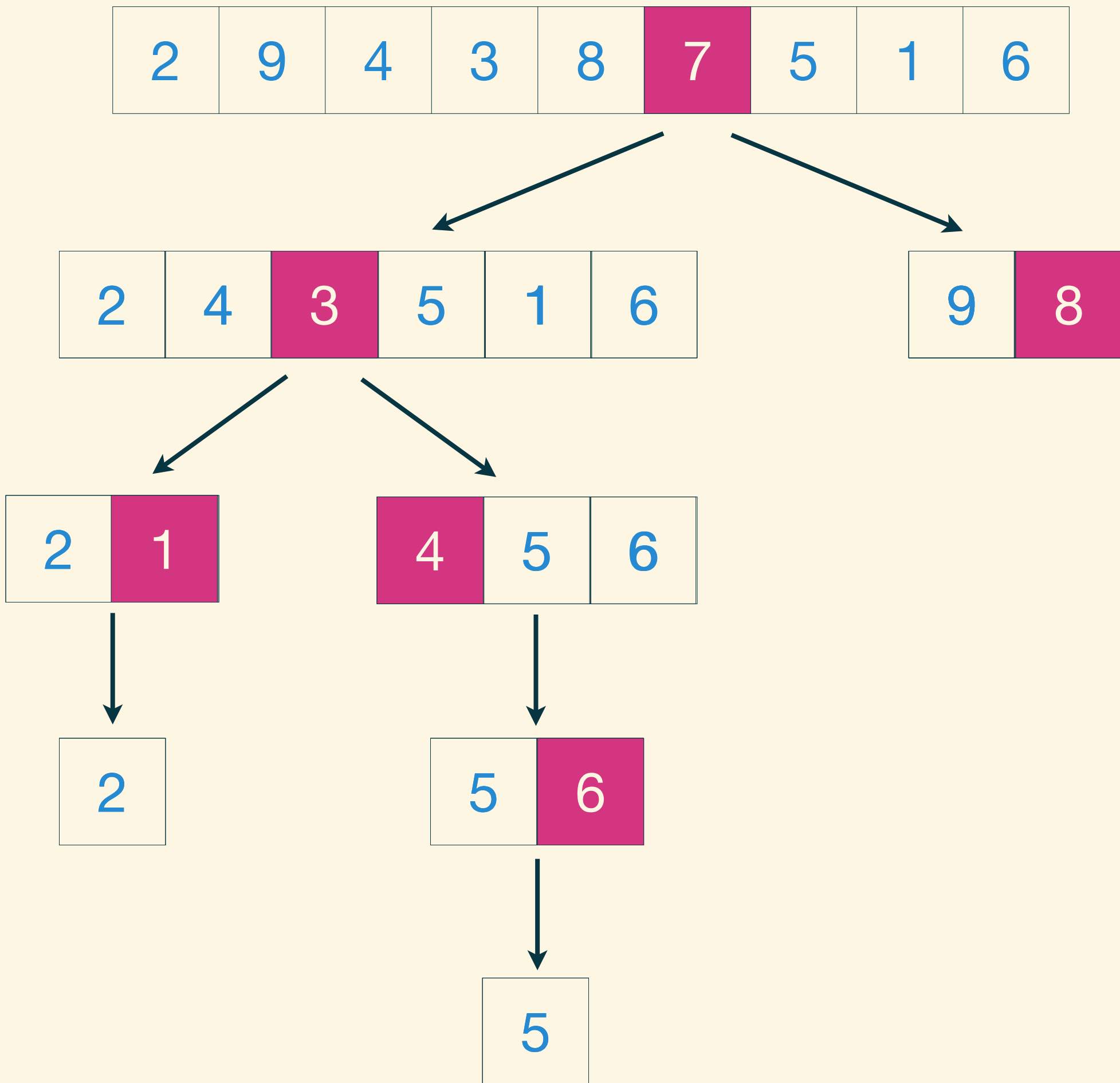


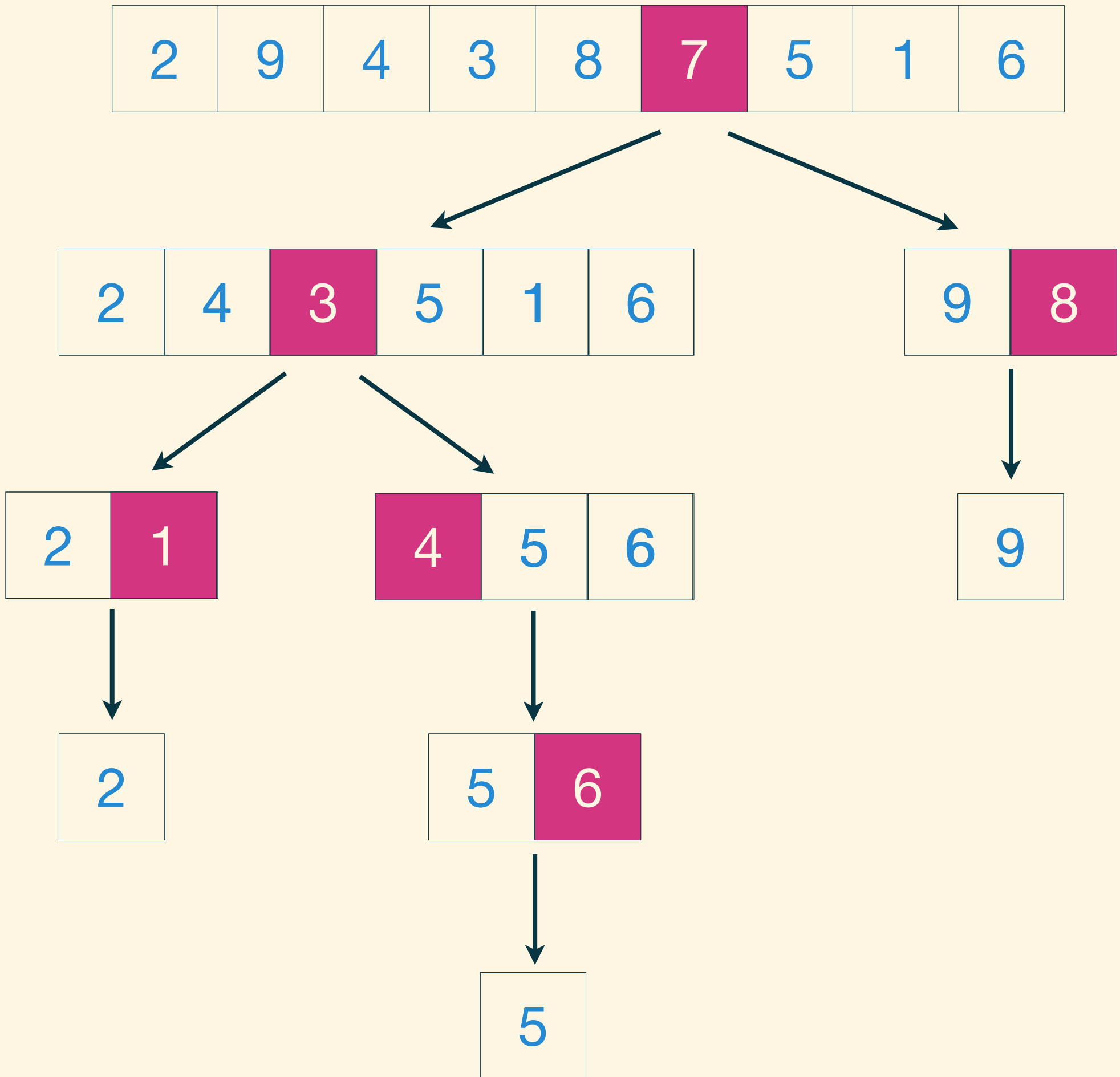










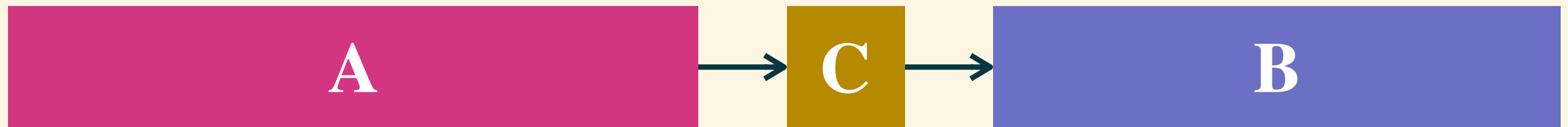


Conquer

由於子數列已被 pivot 分為兩段, 一段
小於等於 pivot 的數列 A, 另一段大於
等於 pivot 的數列 B。

Conquer

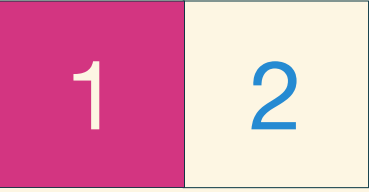
由於子數列已被 pivot 分為兩段, 一段小於等於 pivot 的數列 A, 另一段大於等於 pivot 的數列 B。

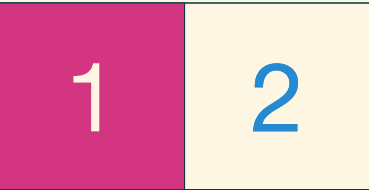


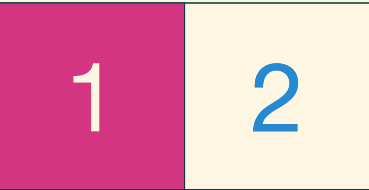
2

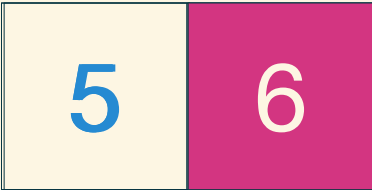
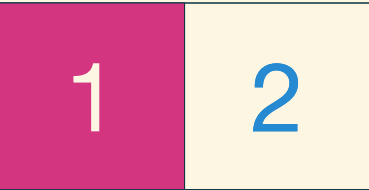
1

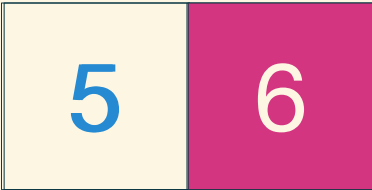
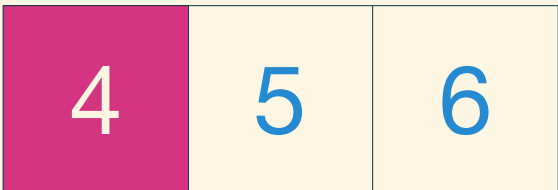
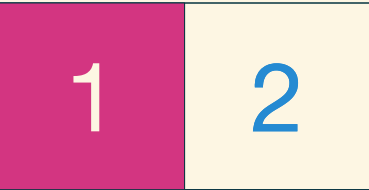
2











3

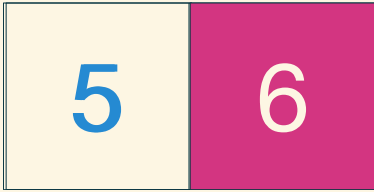
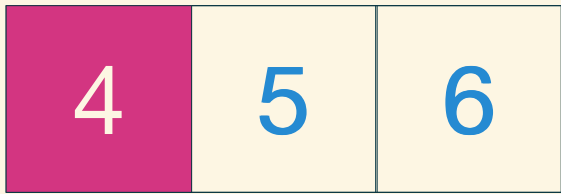
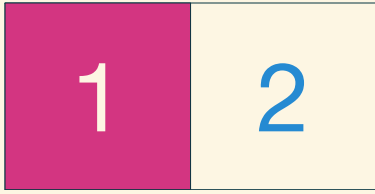
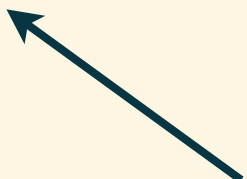
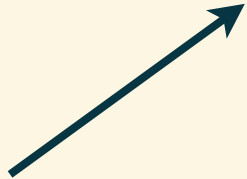
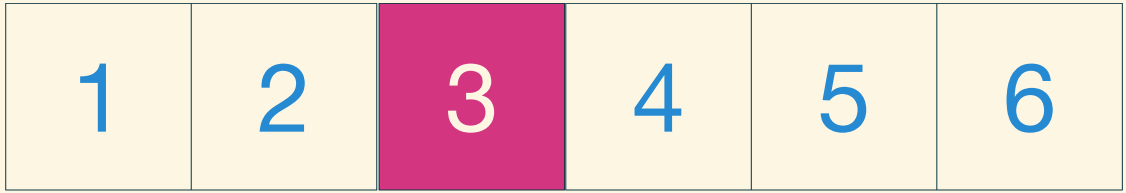
1	2
---	---

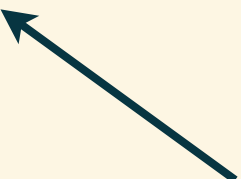
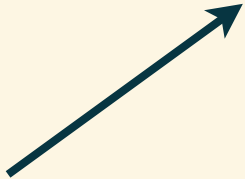
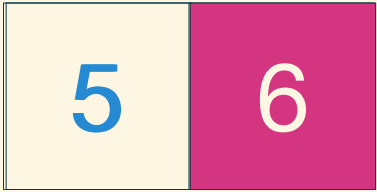
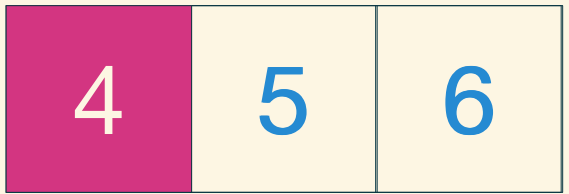
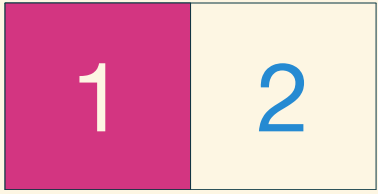
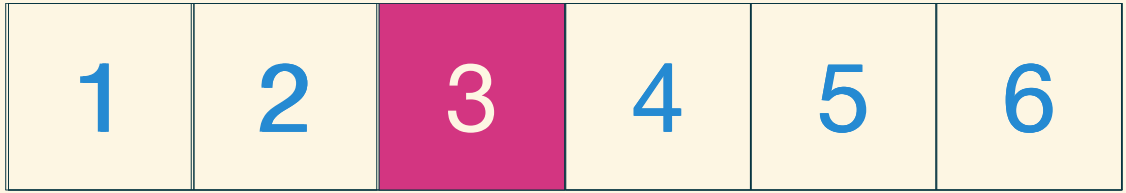
4	5	6
---	---	---

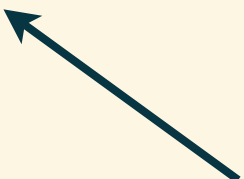
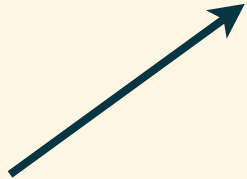
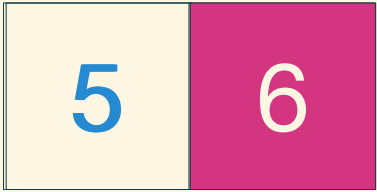
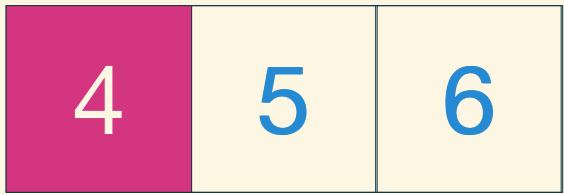
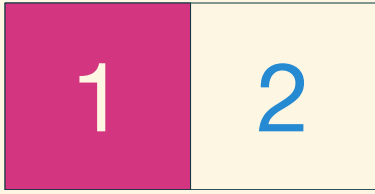
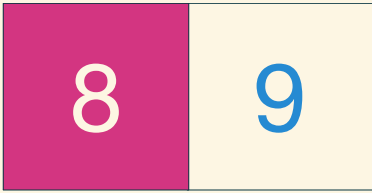
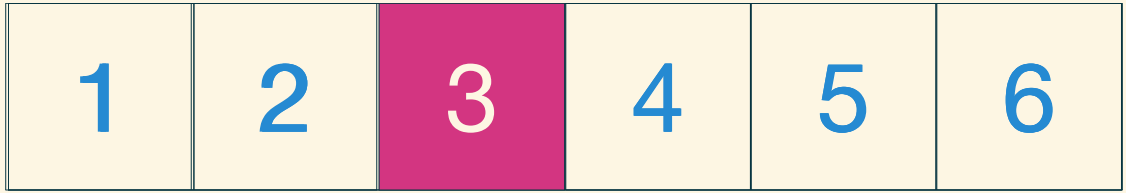
2

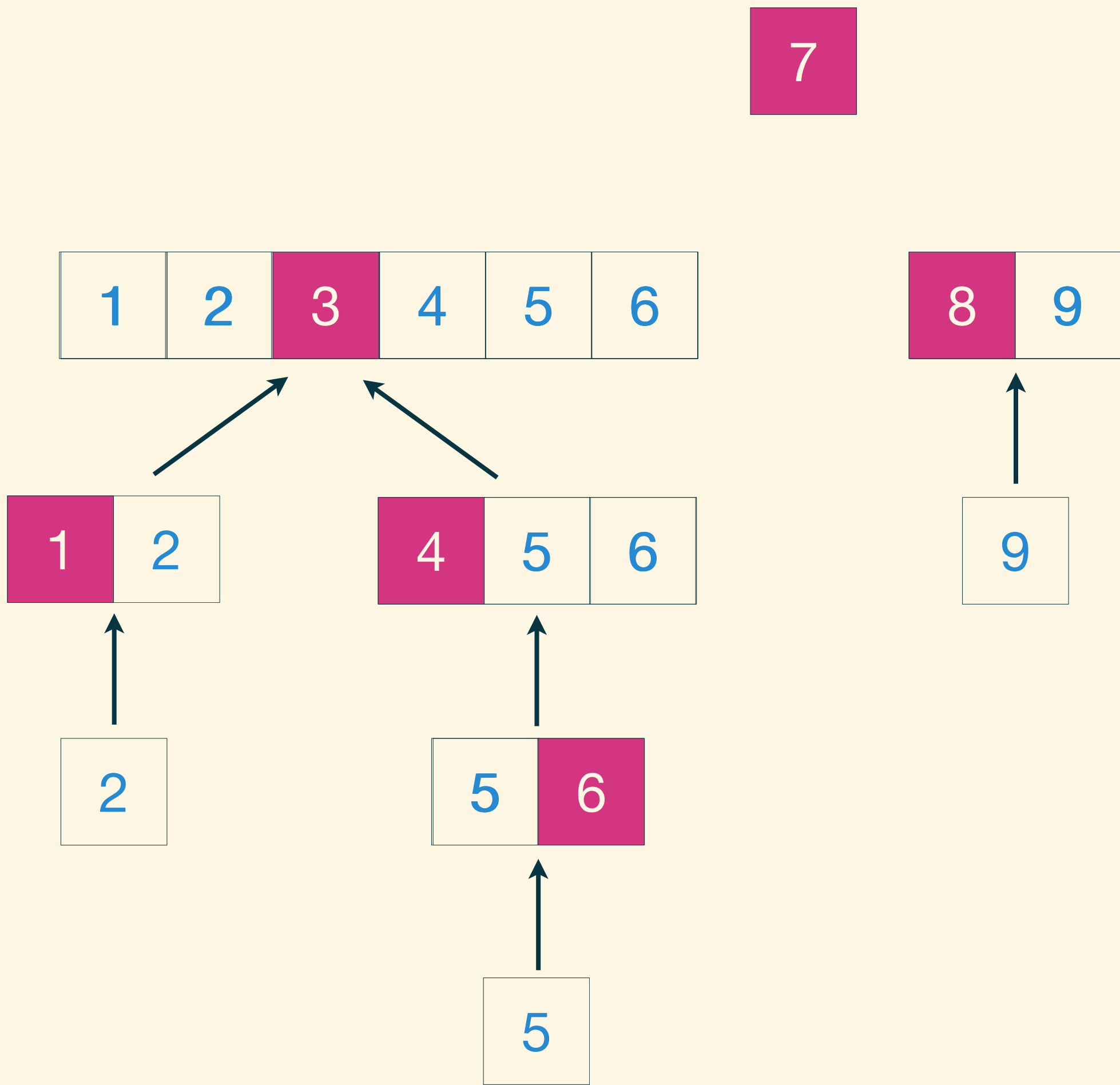
5	6
---	---

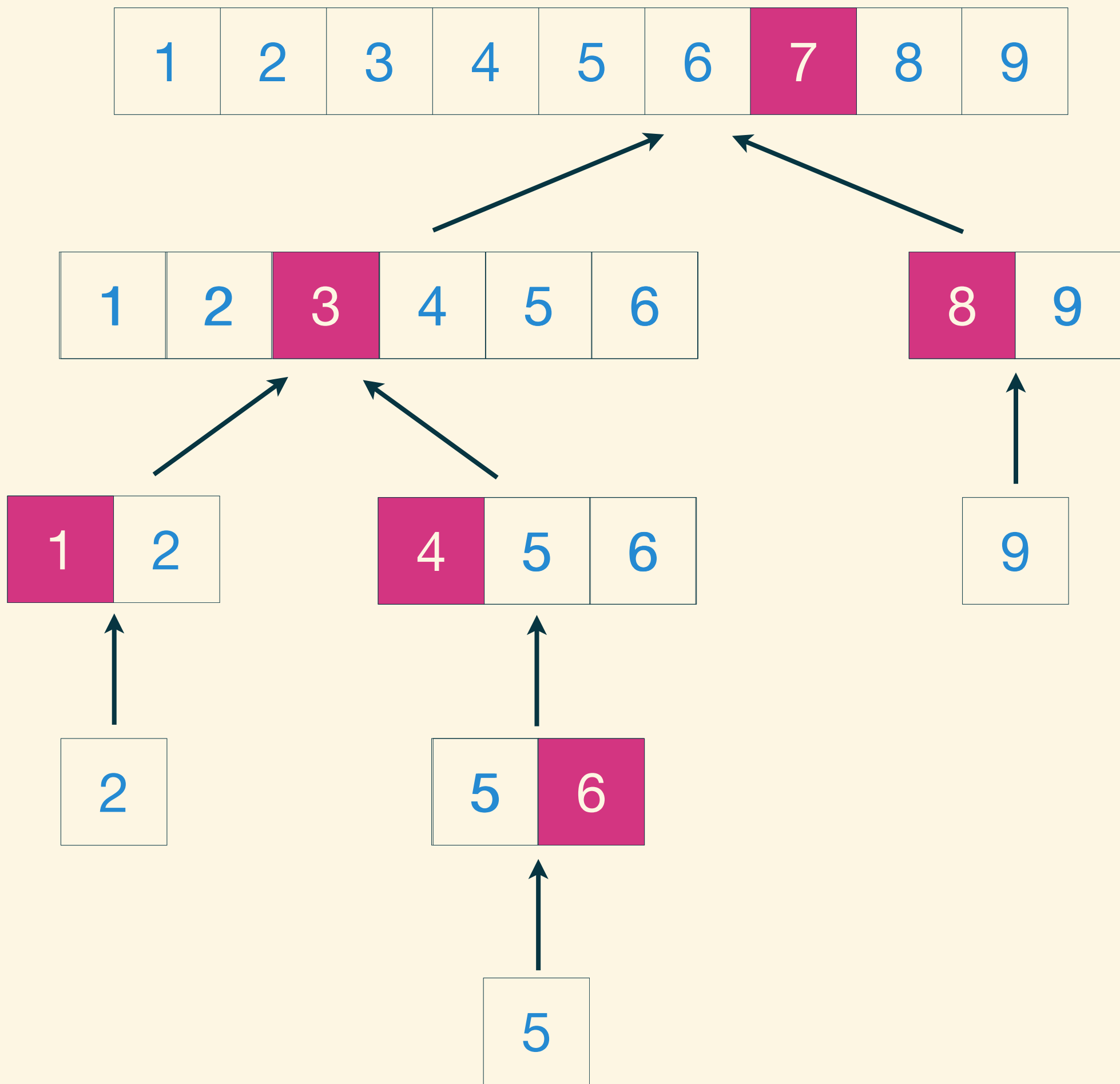
5

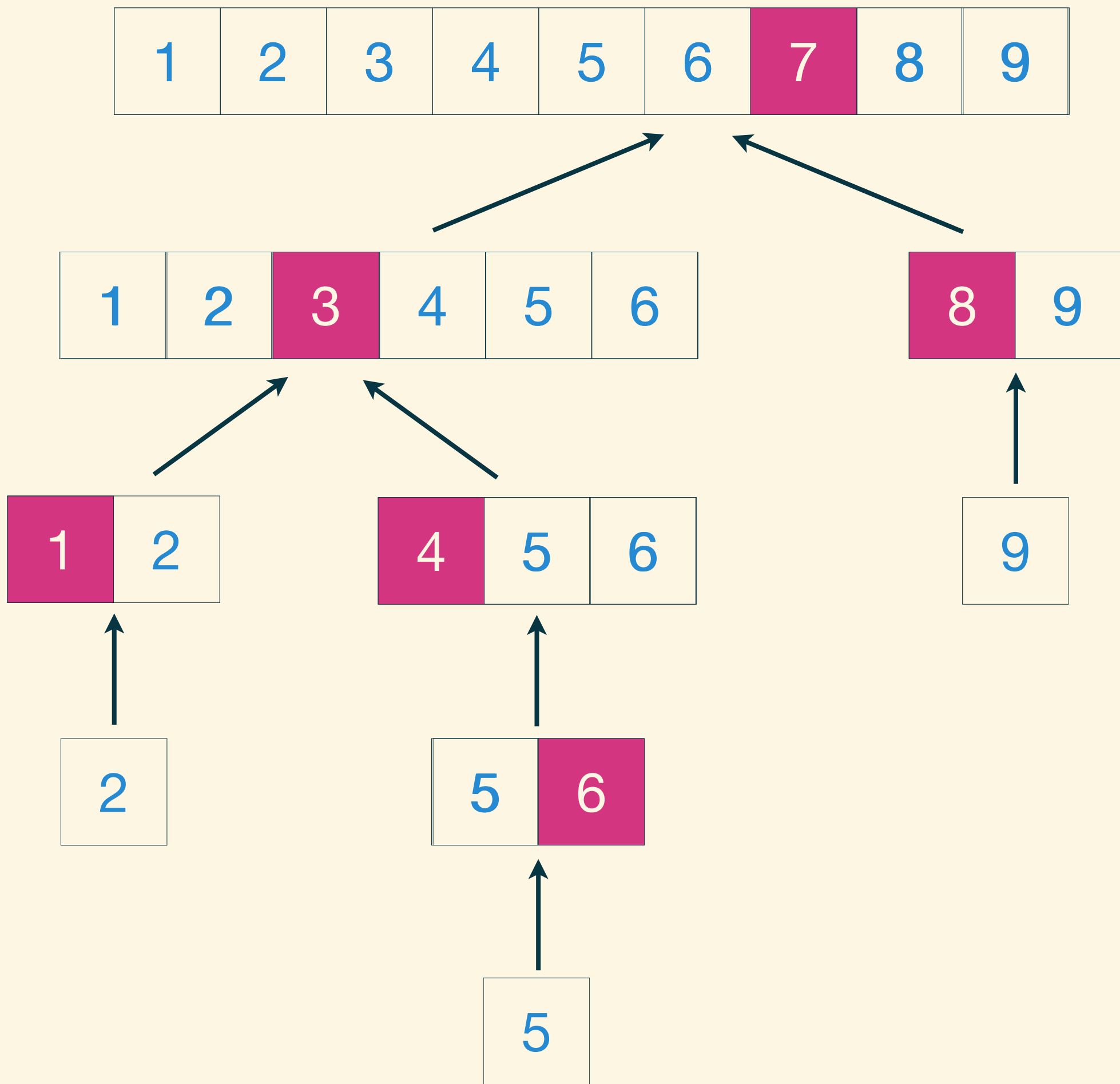








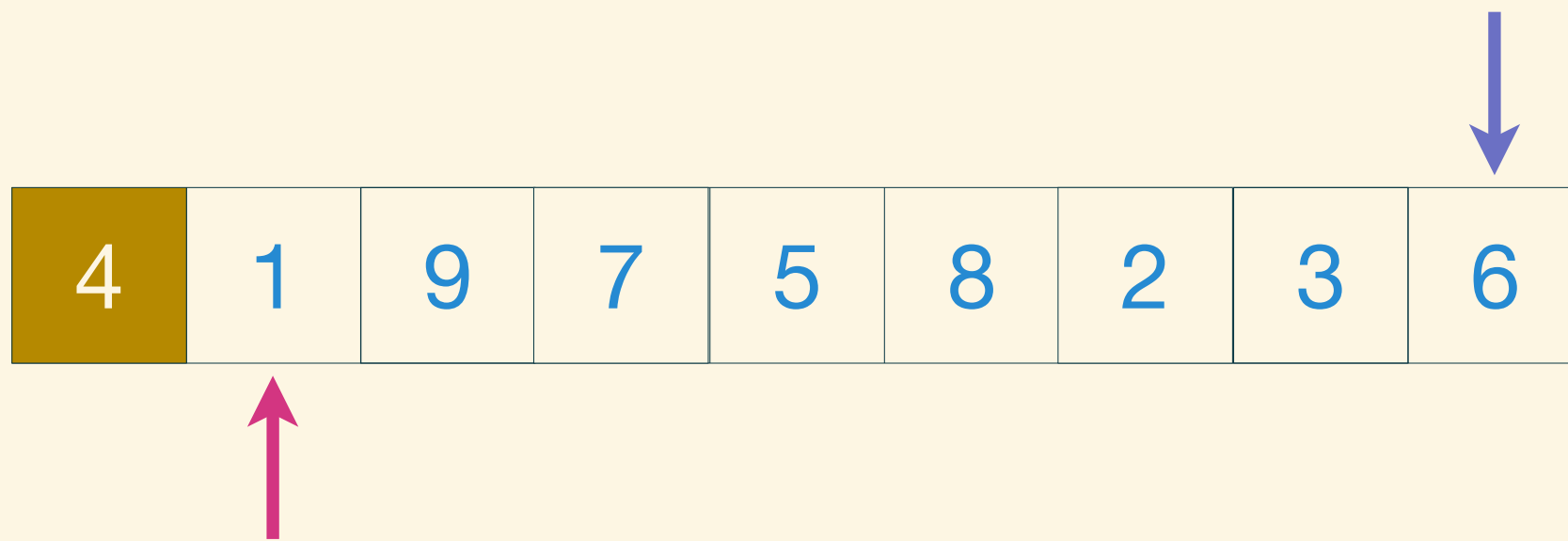




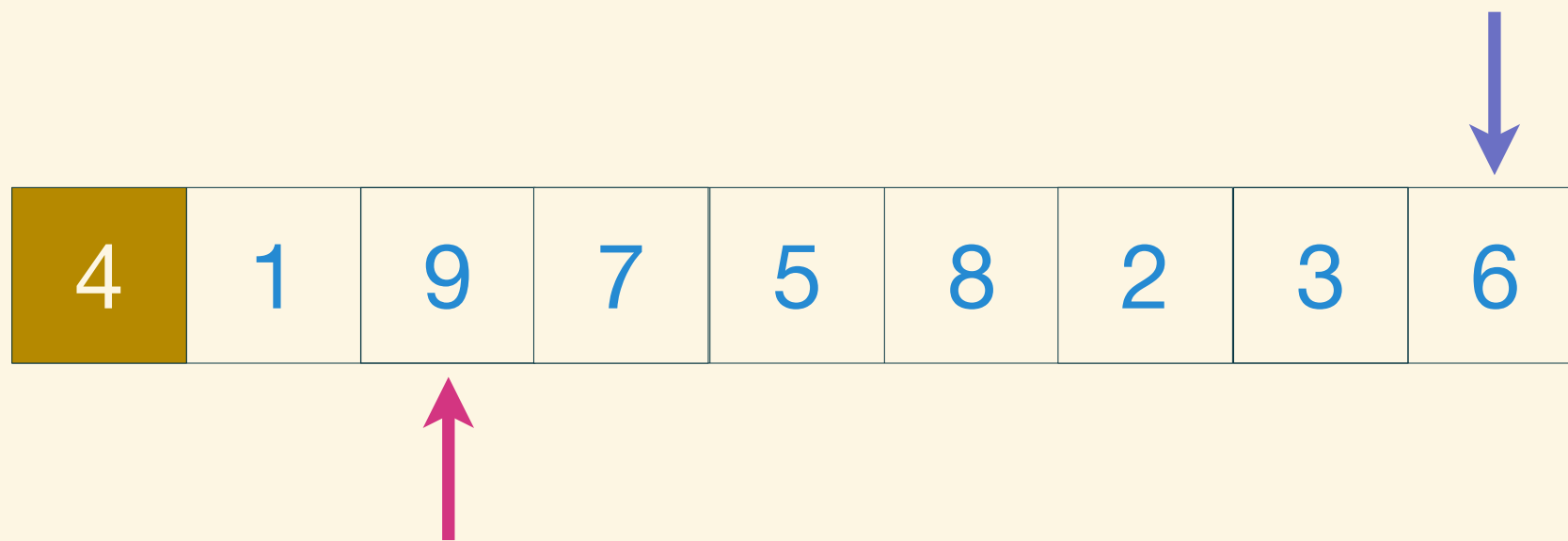
How to Divide

4	1	9	7	5	8	2	3	6
---	---	---	---	---	---	---	---	---

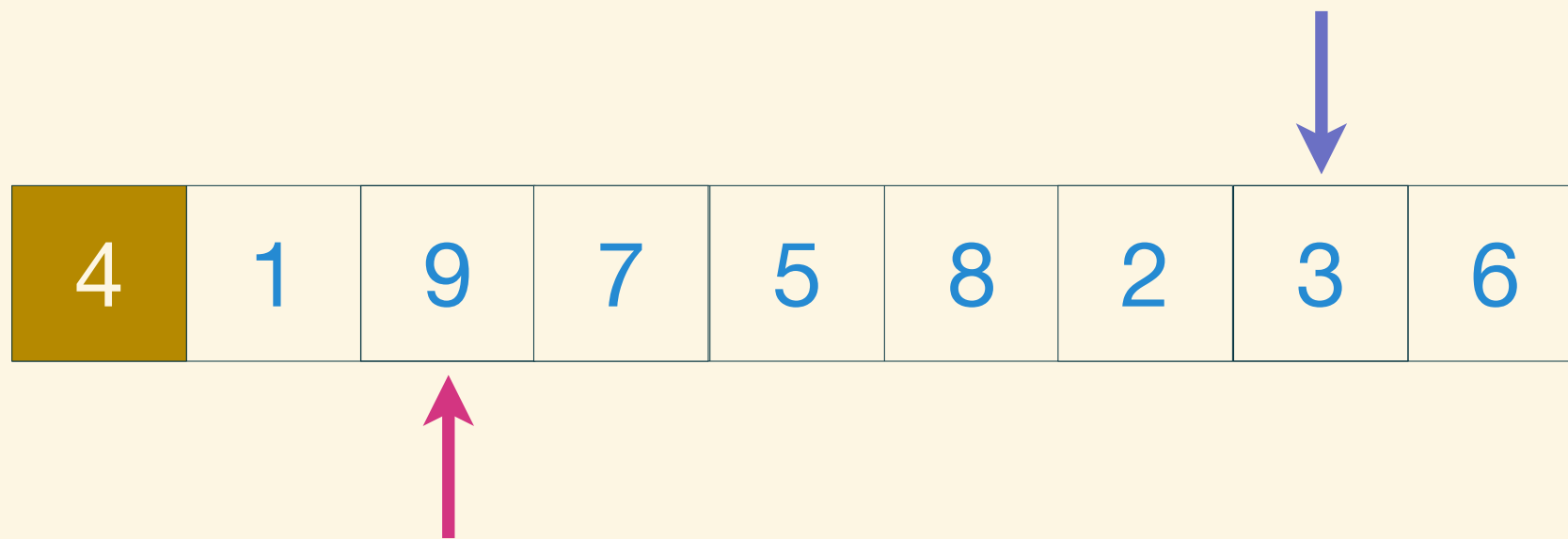
How to Divide



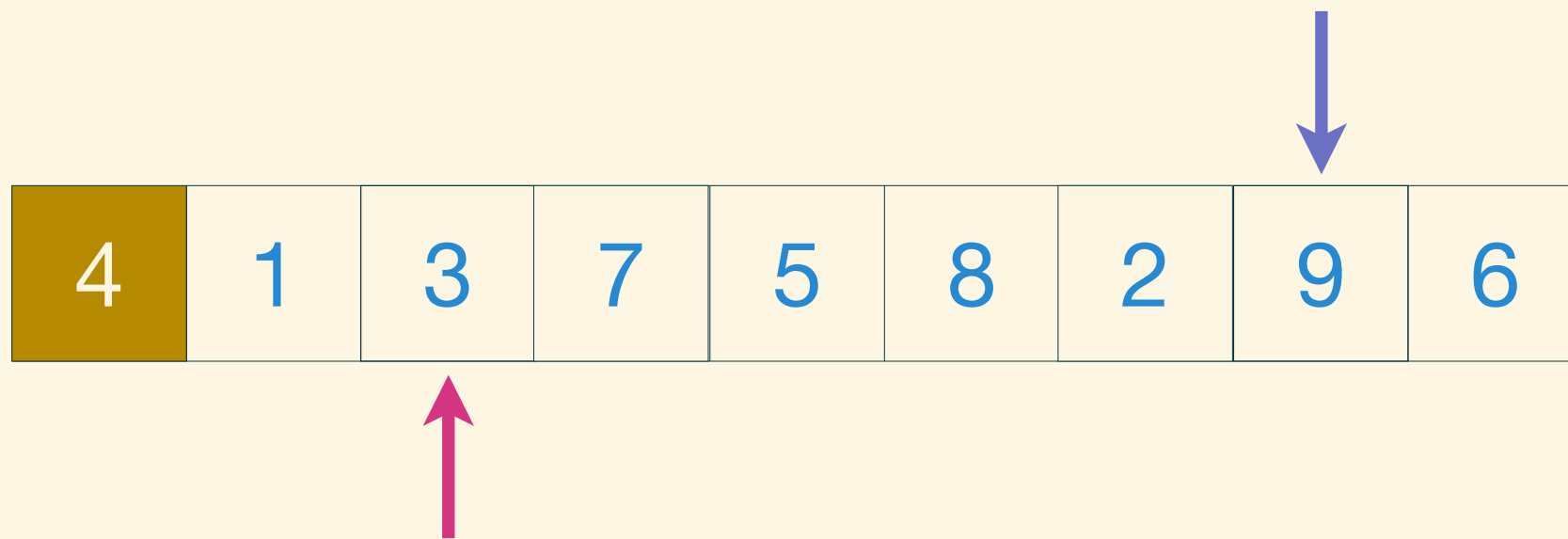
How to Divide



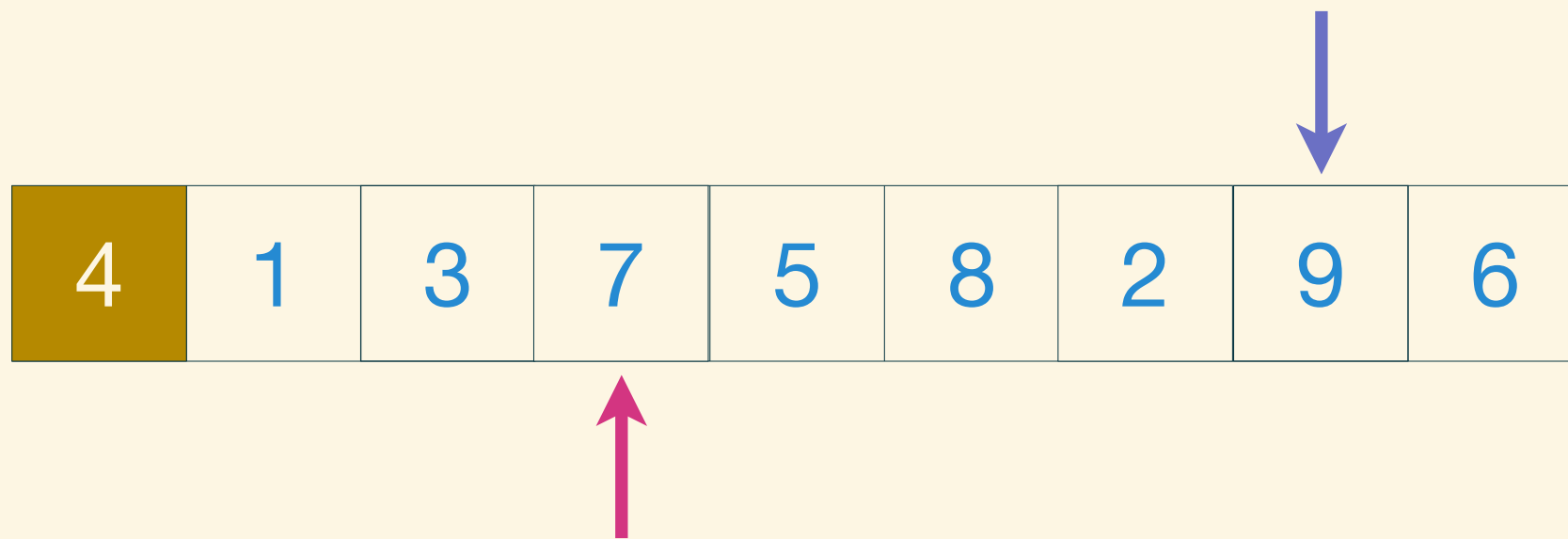
How to Divide



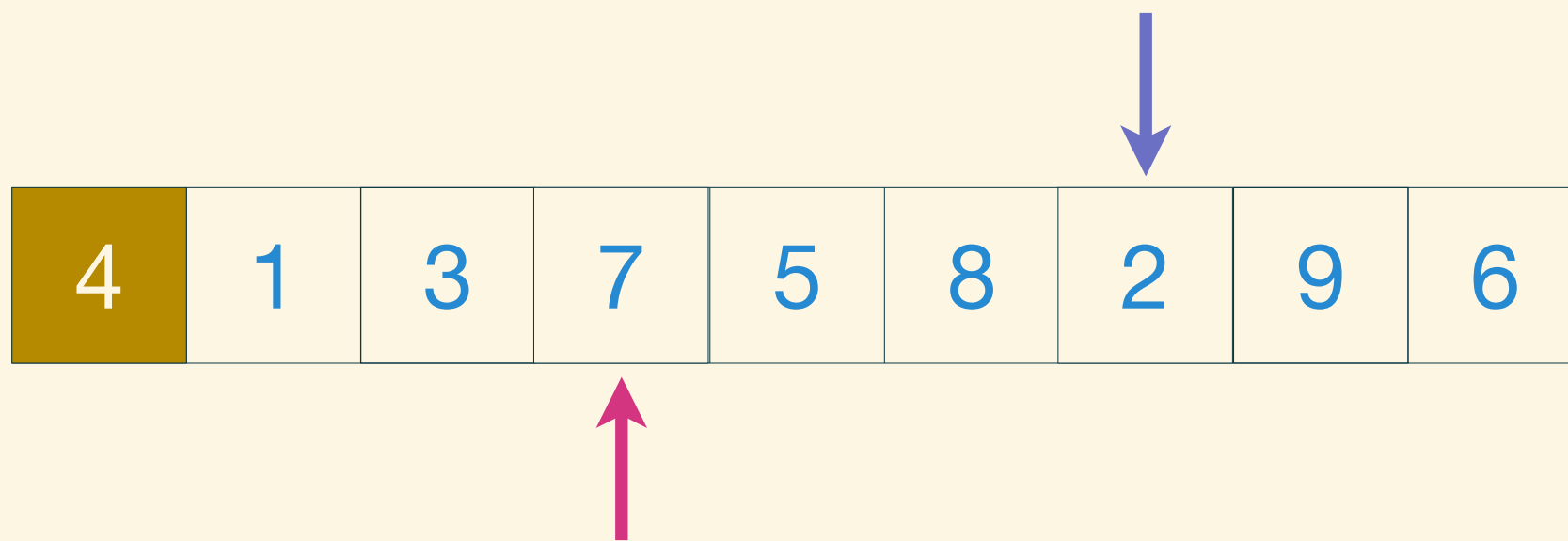
How to Divide



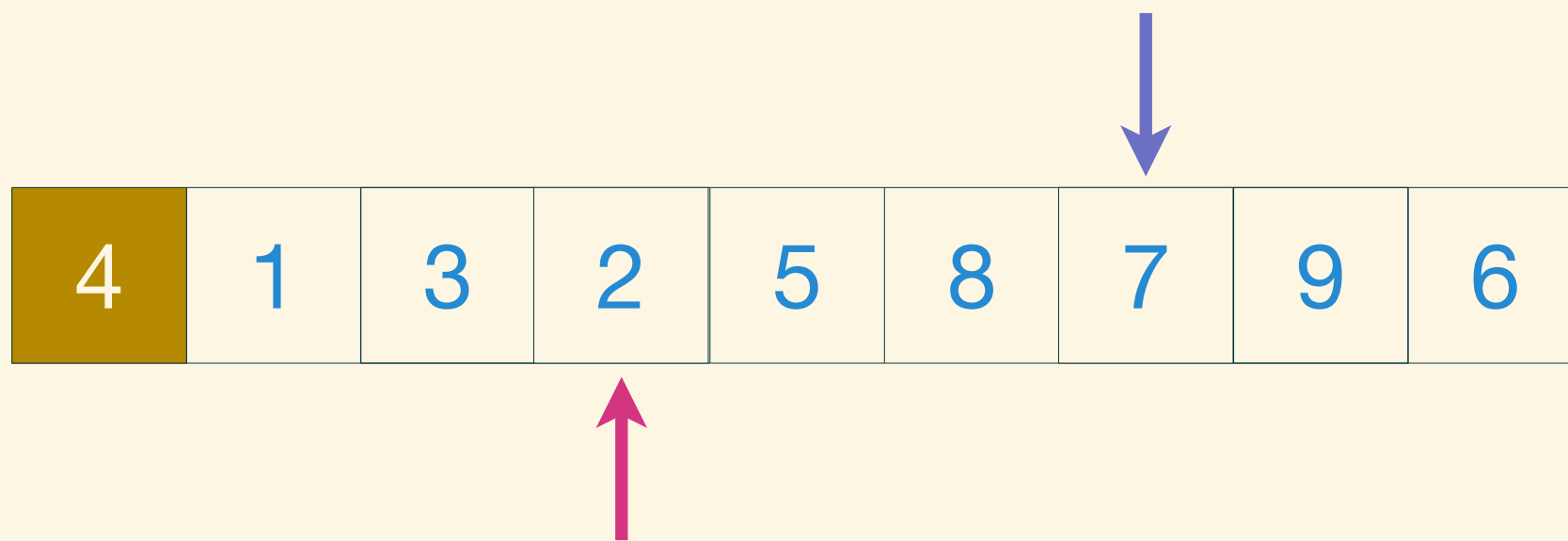
How to Divide



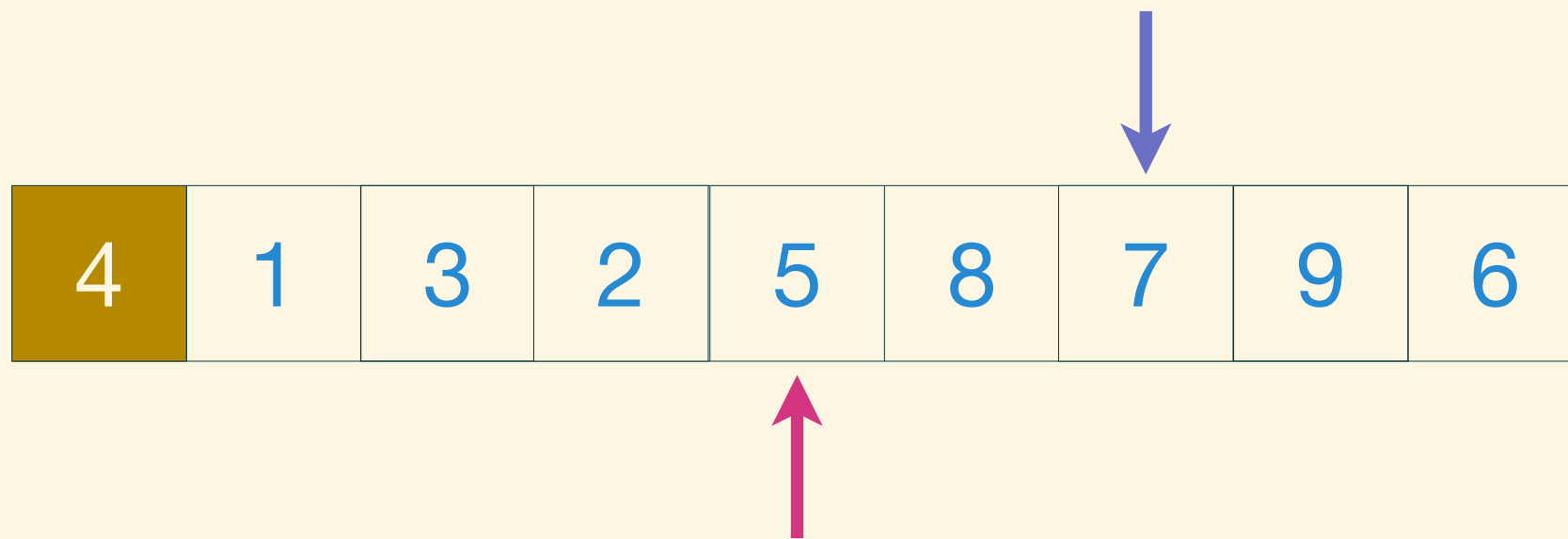
How to Divide



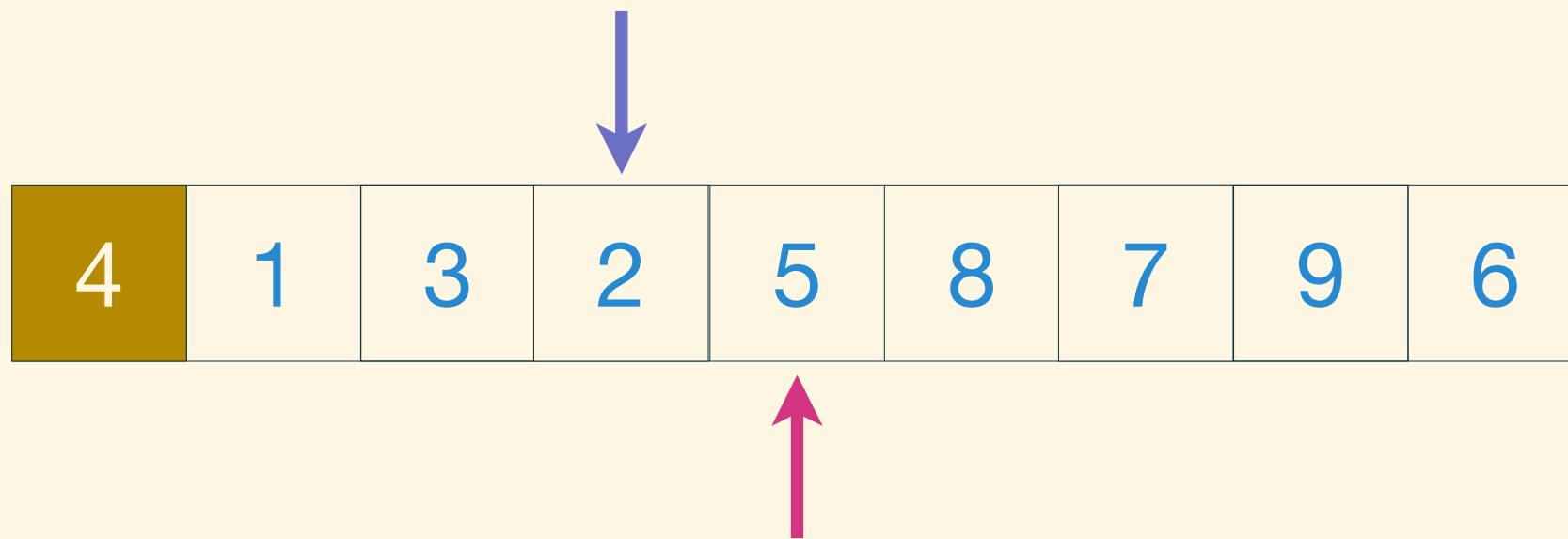
How to Divide



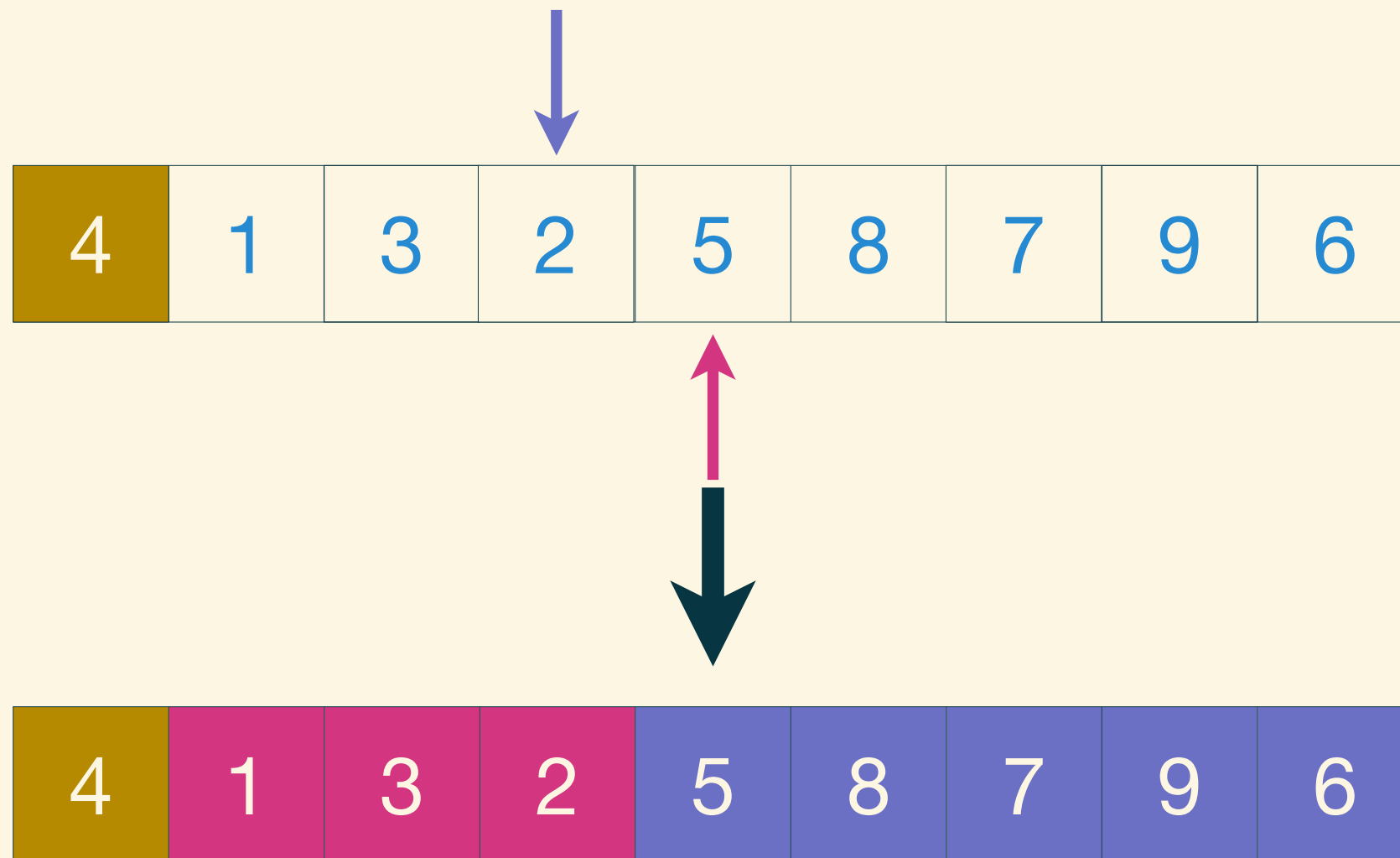
How to Divide



How to Divide



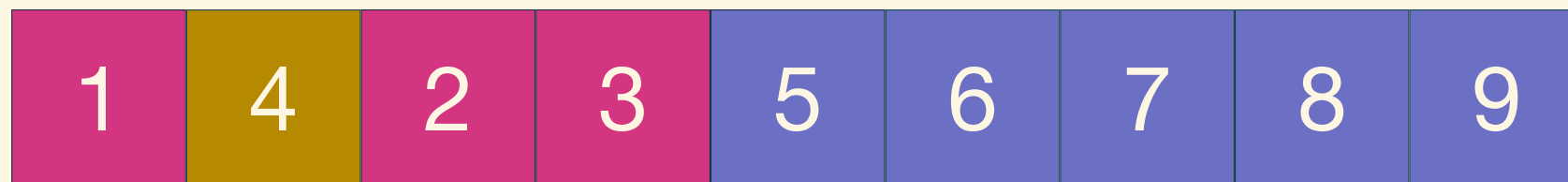
How to Divide



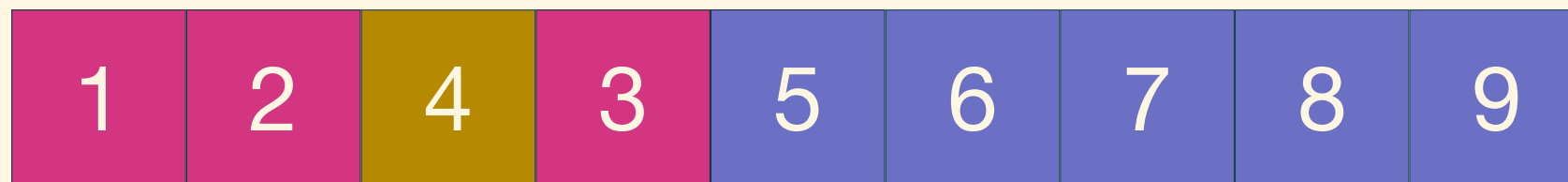
How to Conquer



How to Conquer



How to Conquer



How to Conquer



In-place Version

```
void quickSort( int L, int R ) {  
    if ( R - L <= 1 )  
        return;  
    int pivot = N[ R - 1 ], p = L;  
    for ( int i = L; i < R - 1; ++i ) {  
        if ( N[ i ] <= pivot ) {  
            swap( N[ i ], N[ p ] );  
            ++p;  
        }  
    }  
    swap( N[ R - 1 ], N[ p ] );  
    quickSort( L, p );  
    quickSort( p + 1, R );  
}
```

How to Divide

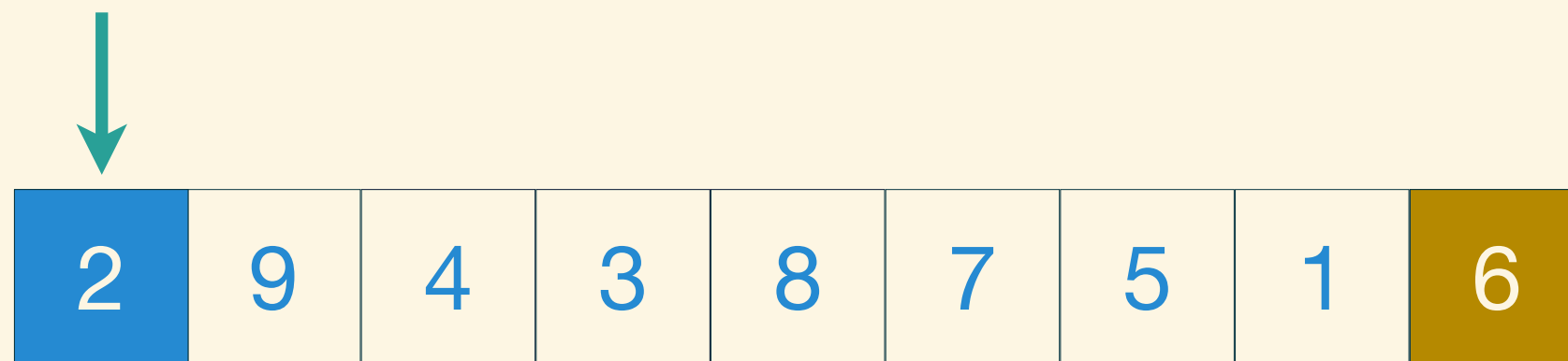
2	9	4	3	8	7	5	1	6
---	---	---	---	---	---	---	---	---

How to Divide

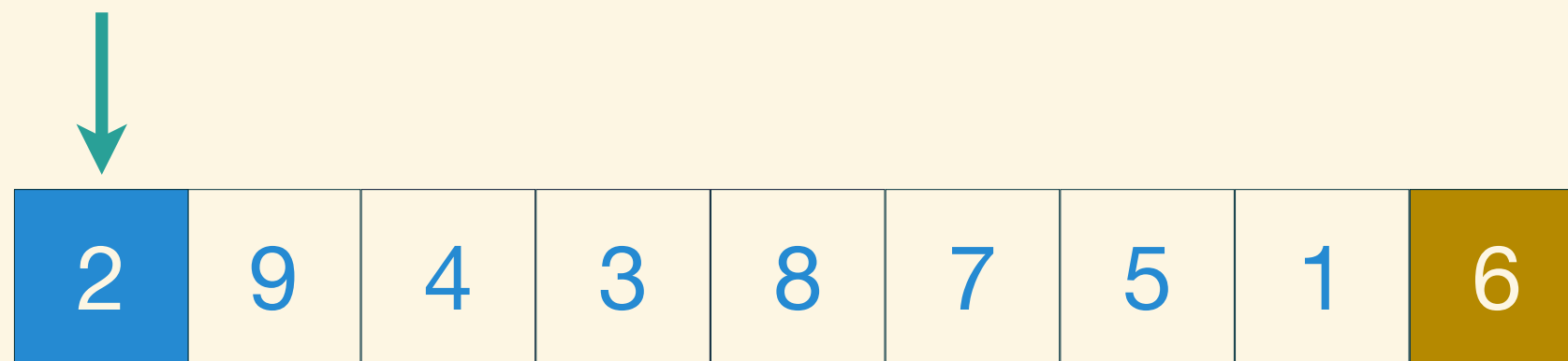


2	9	4	3	8	7	5	1	6
---	---	---	---	---	---	---	---	---

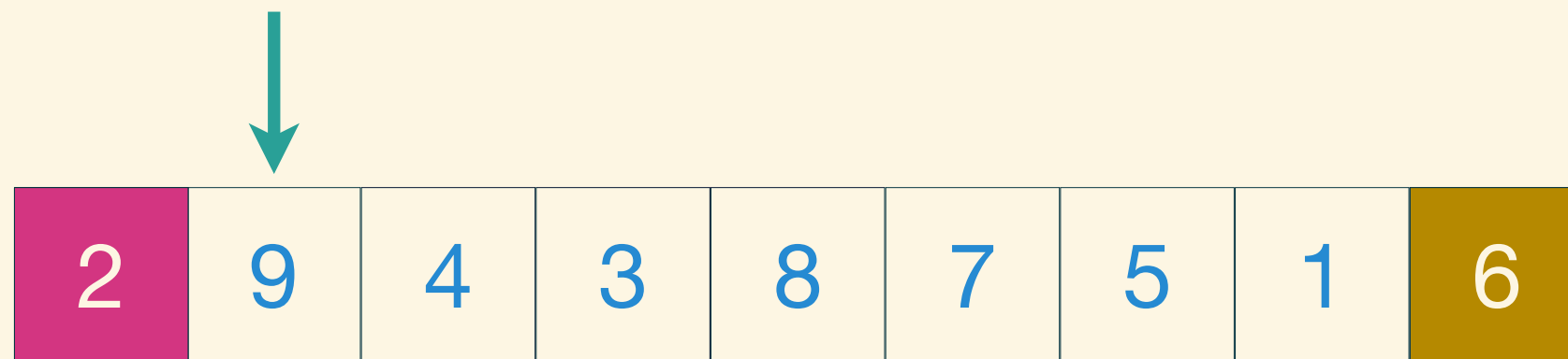
How to Divide



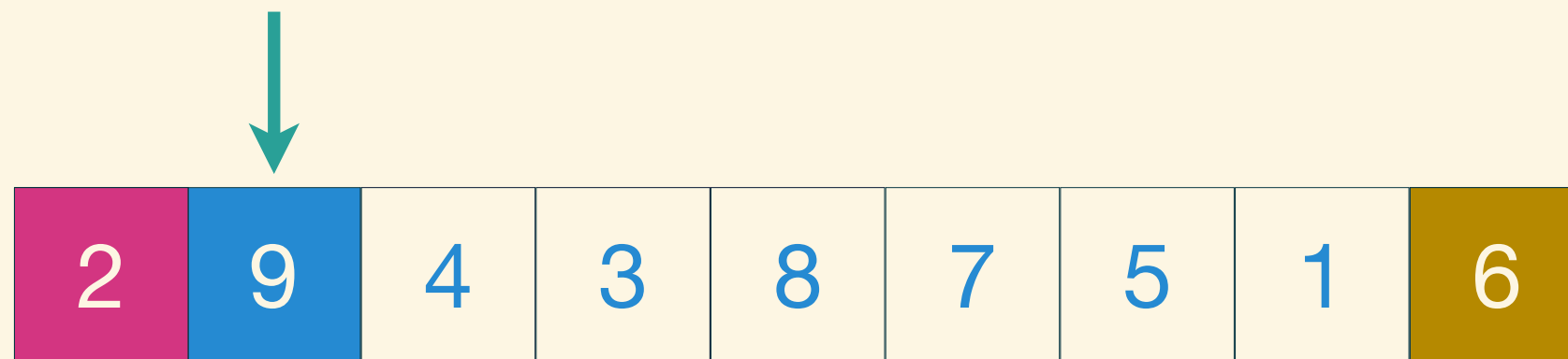
How to Divide



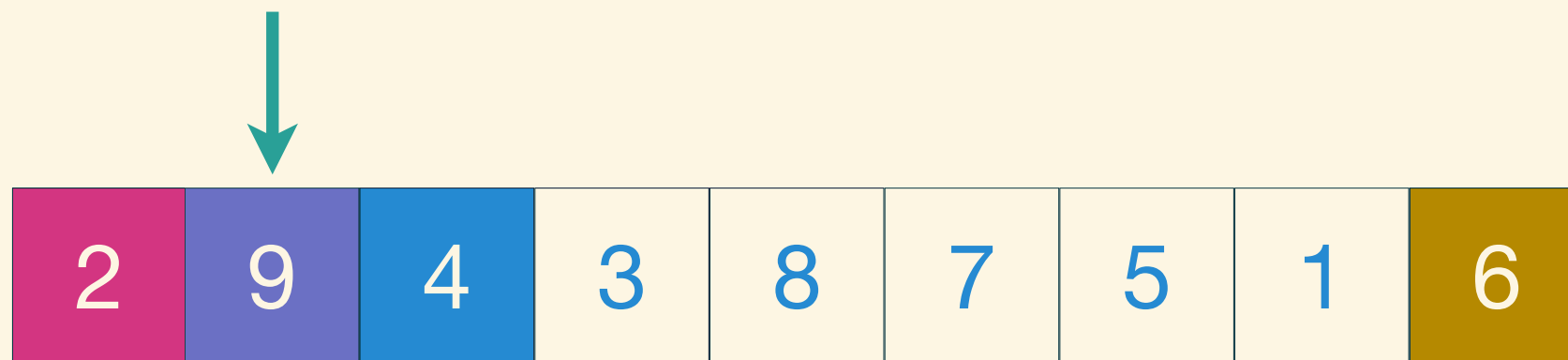
How to Divide



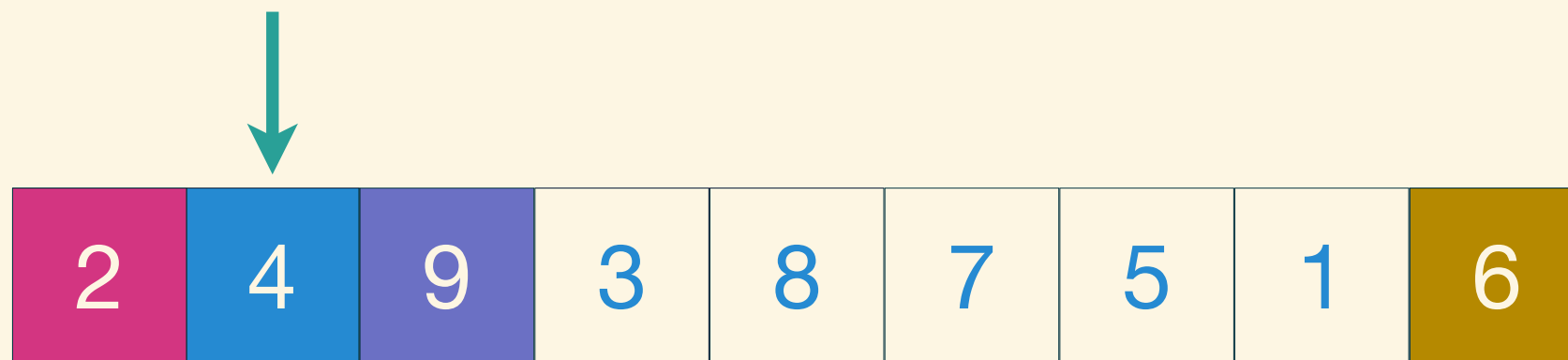
How to Divide



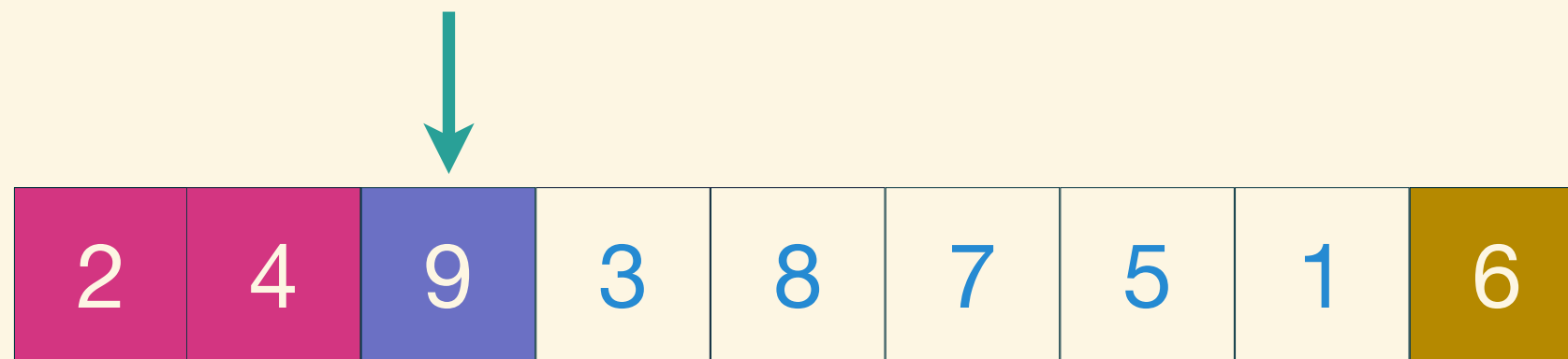
How to Divide



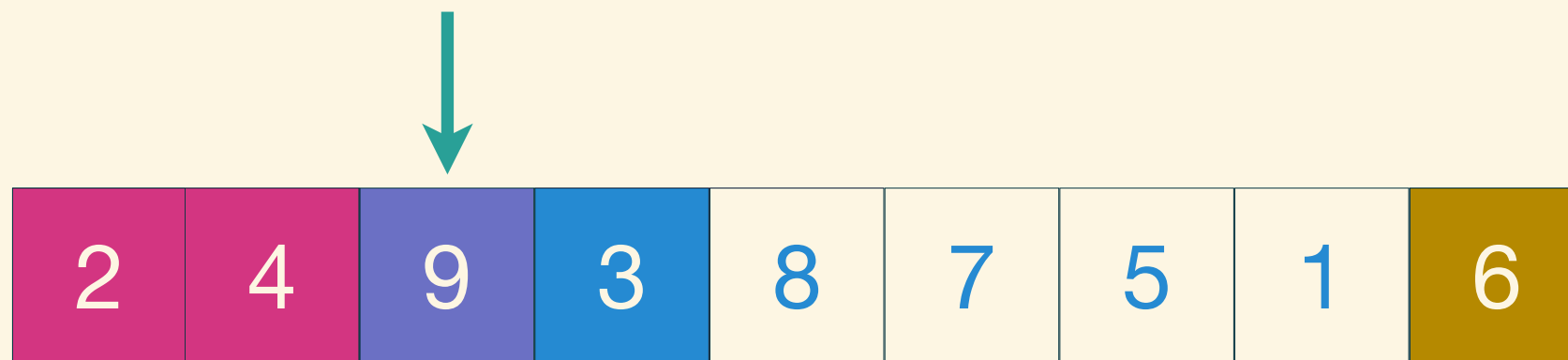
How to Divide



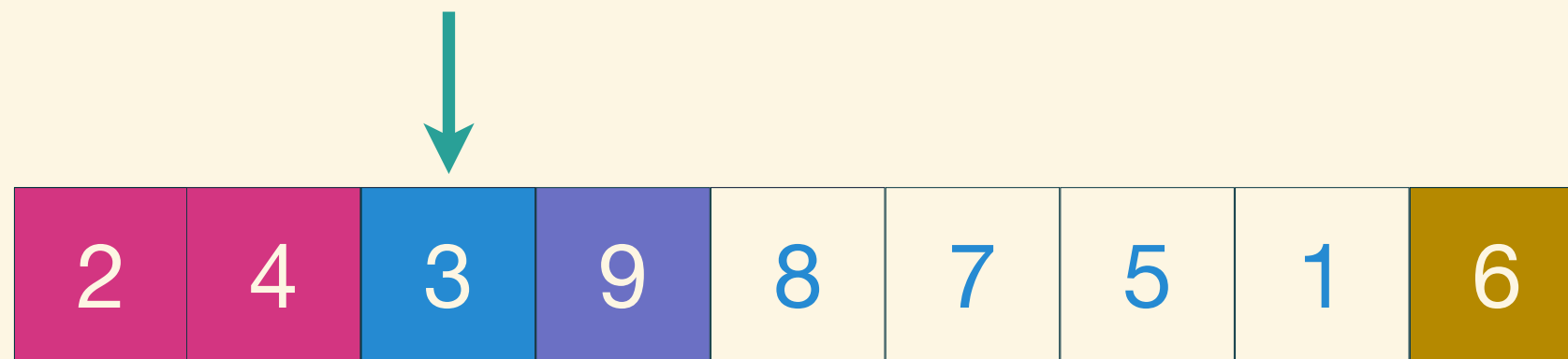
How to Divide



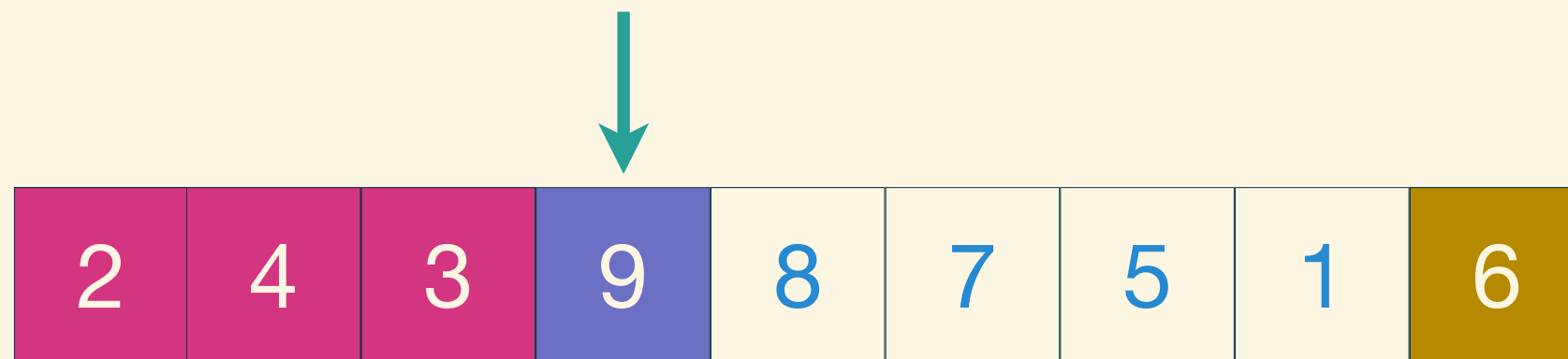
How to Divide



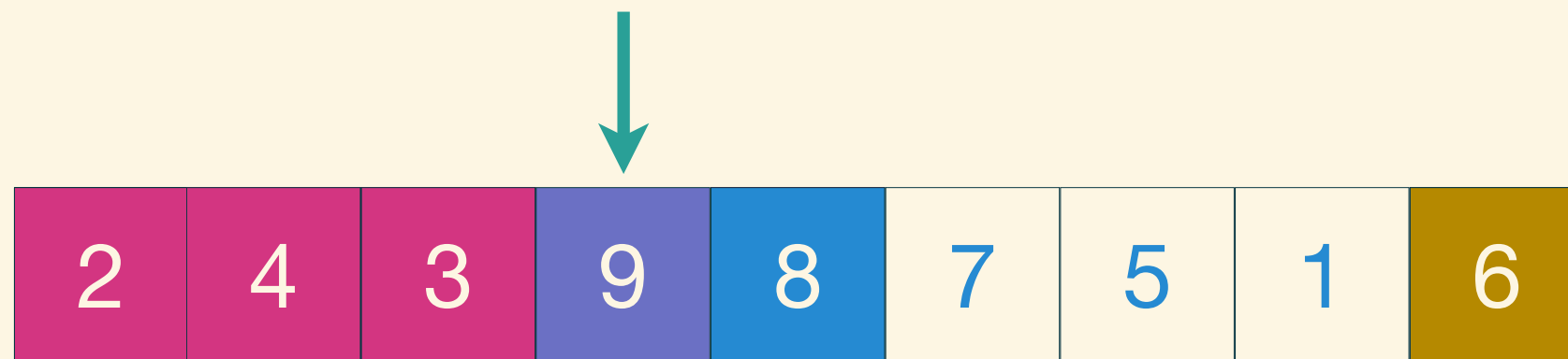
How to Divide



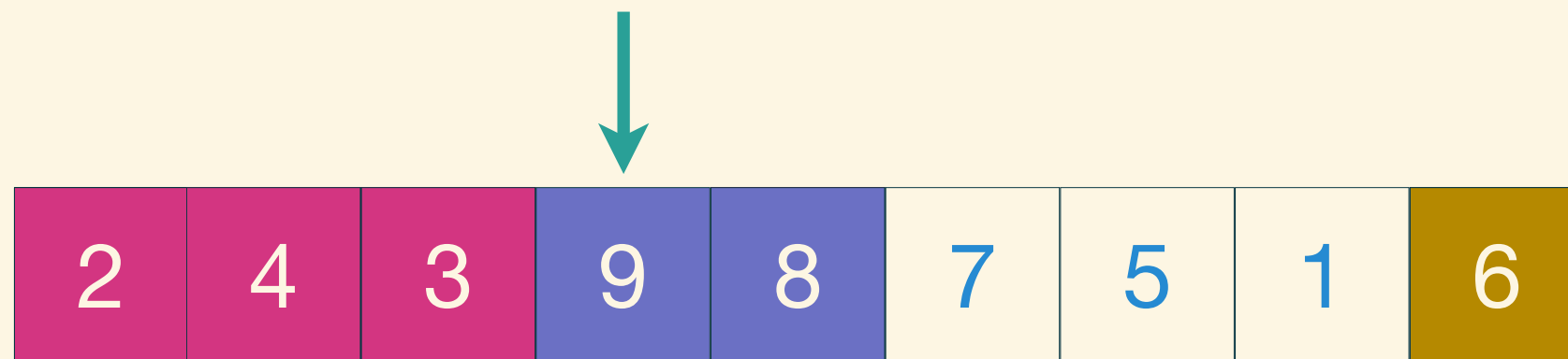
How to Divide



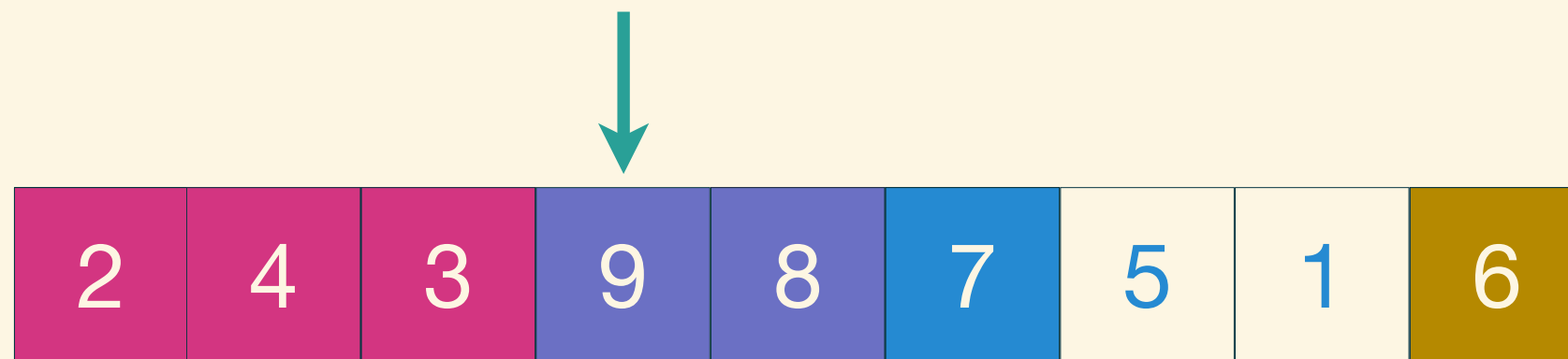
How to Divide



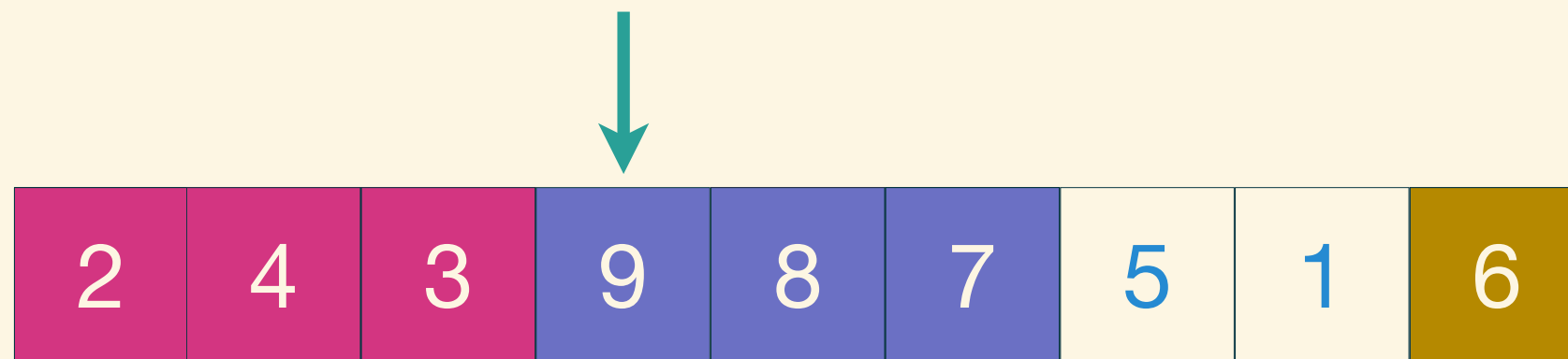
How to Divide



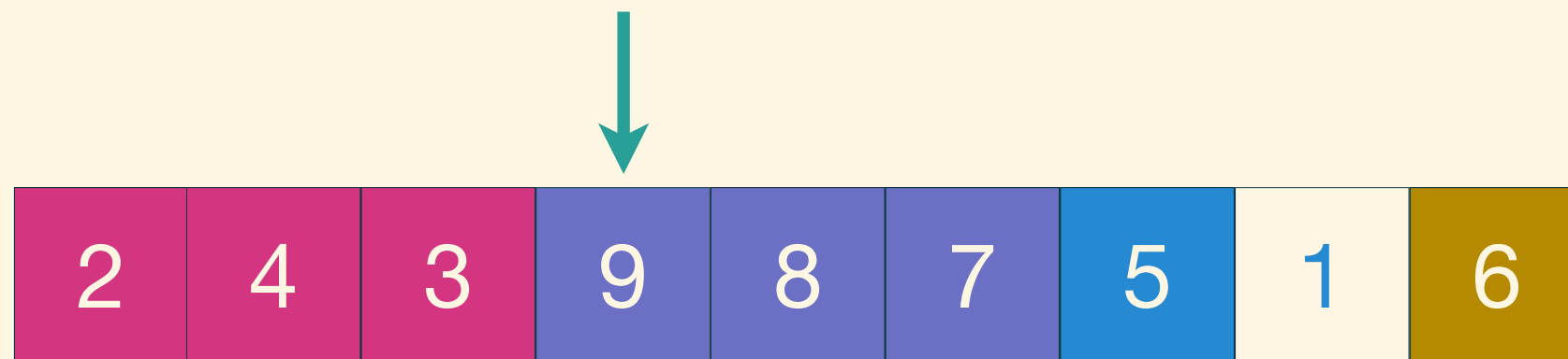
How to Divide



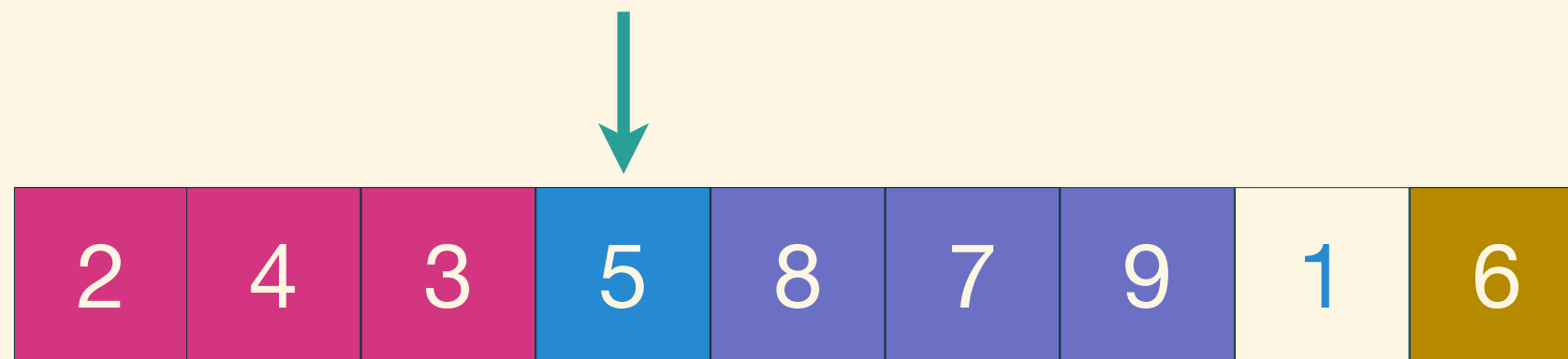
How to Divide



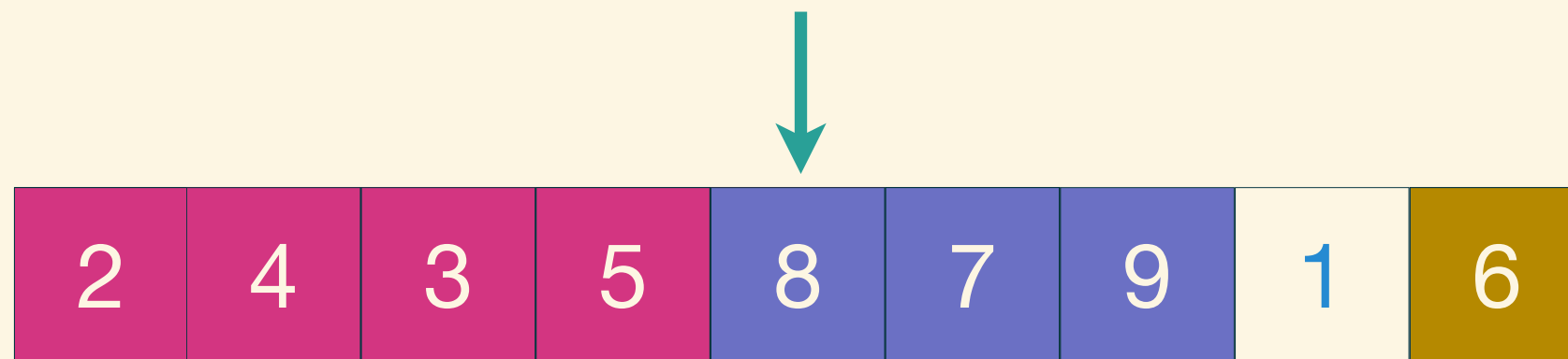
How to Divide



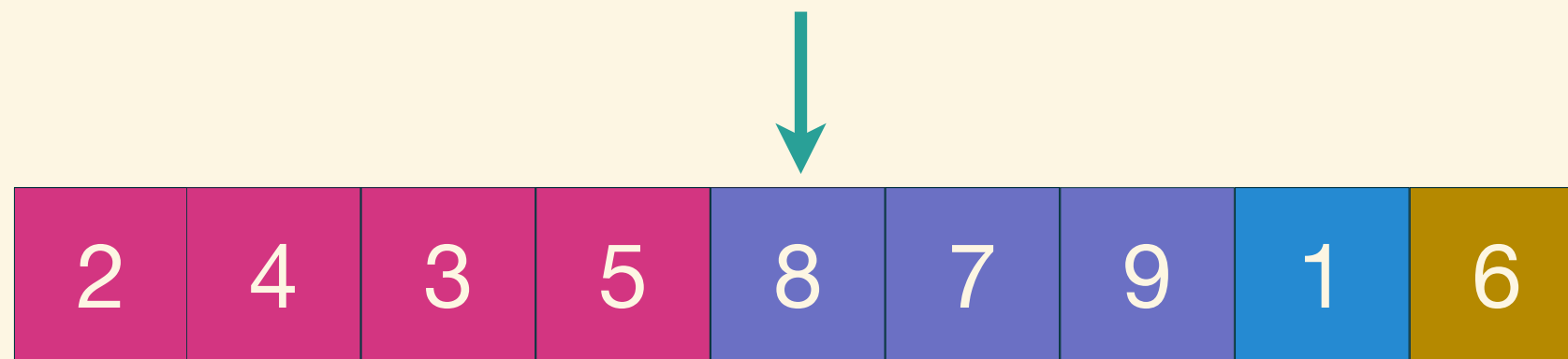
How to Divide



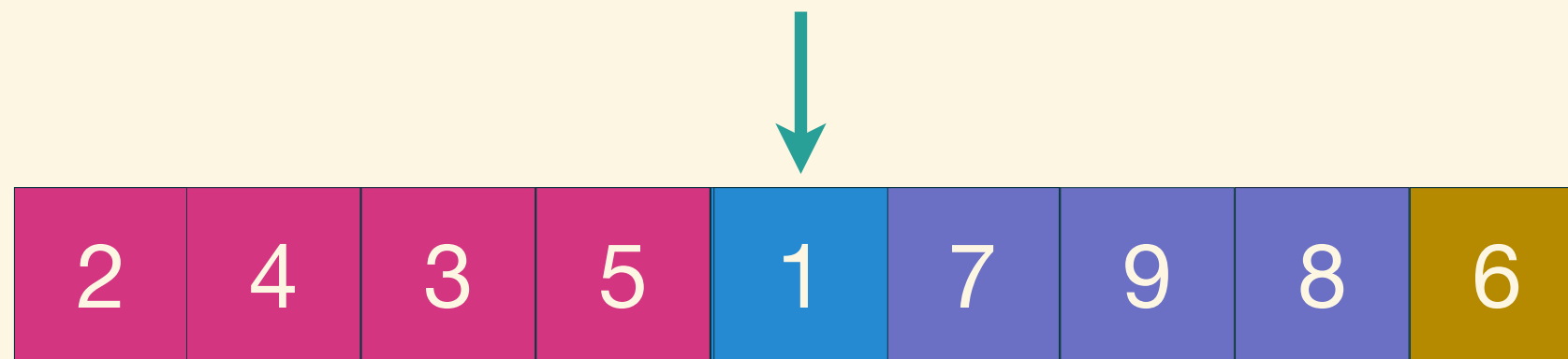
How to Divide



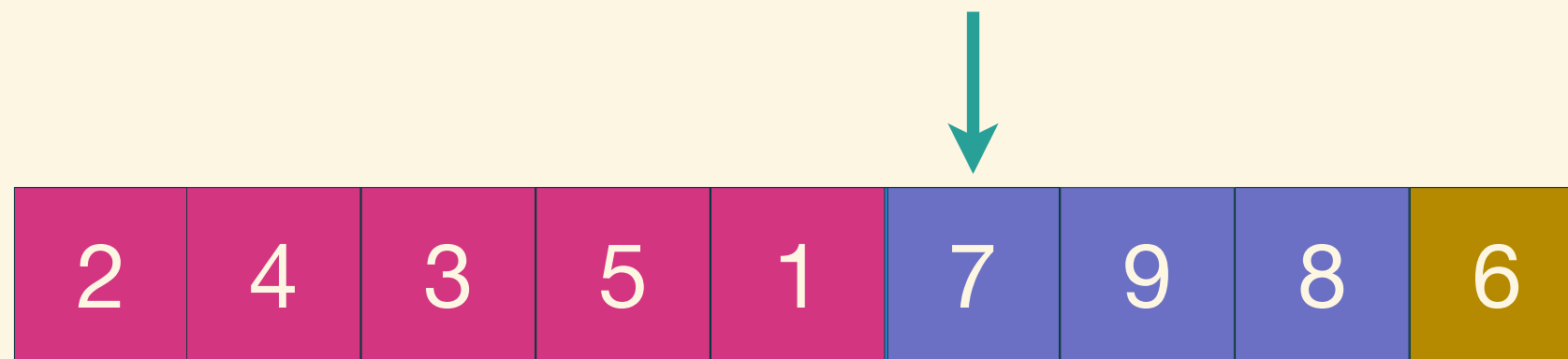
How to Divide



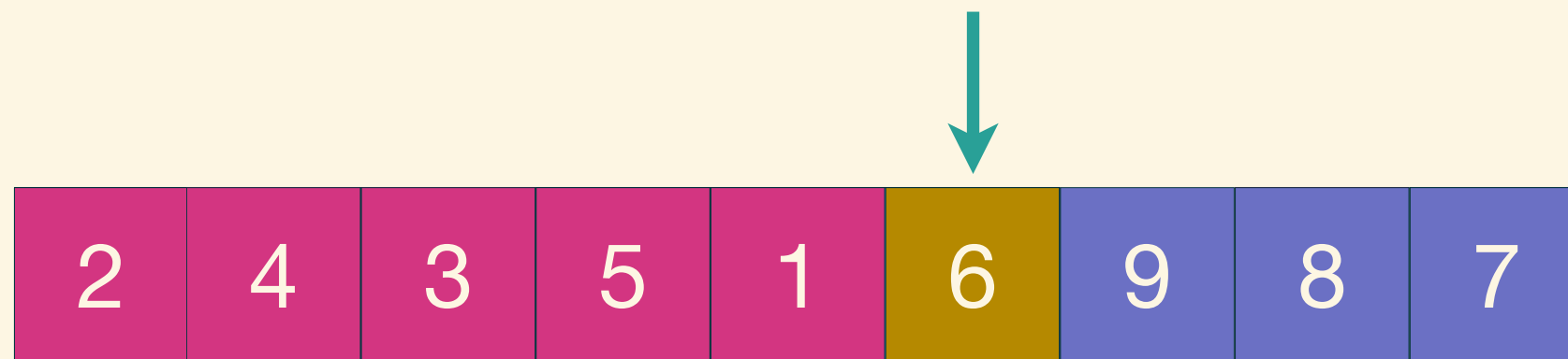
How to Divide



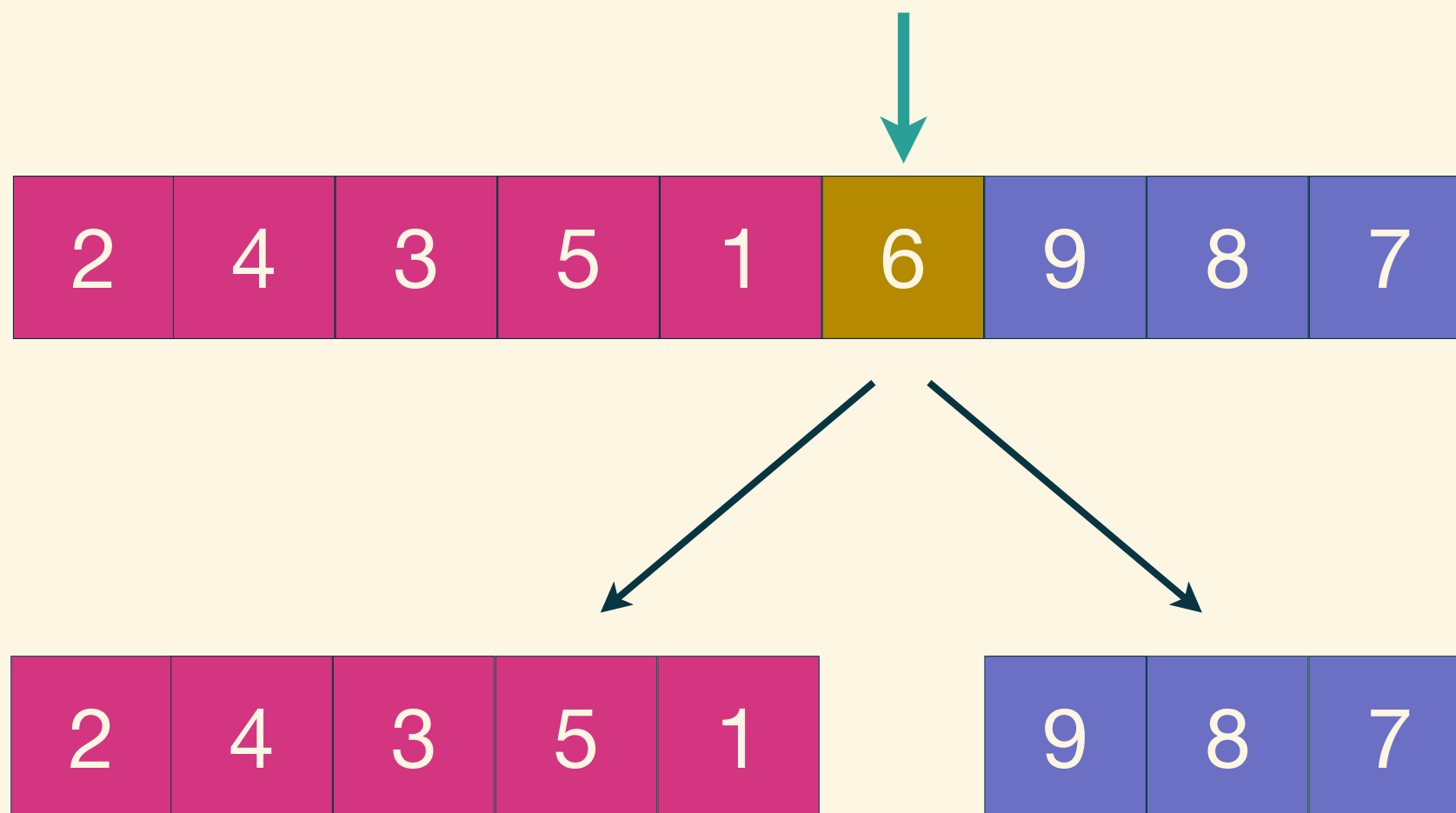
How to Divide



How to Divide



How to Divide



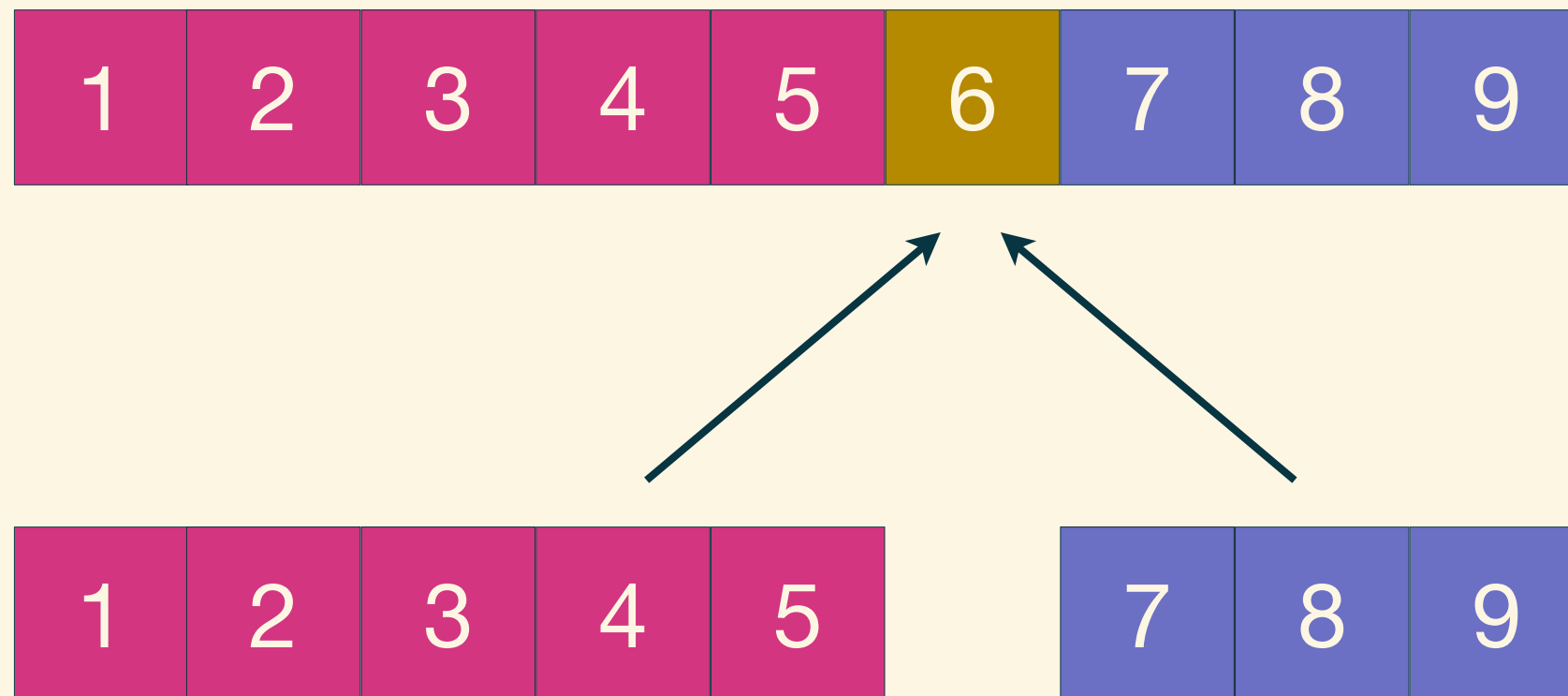
How to Conquer

6

1	2	3	4	5
---	---	---	---	---

7	8	9
---	---	---

How to Conquer



End

