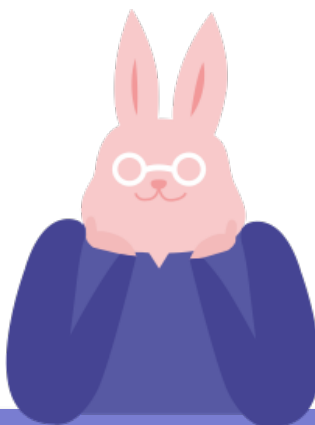


Divide & Conquer, and Greedy approach

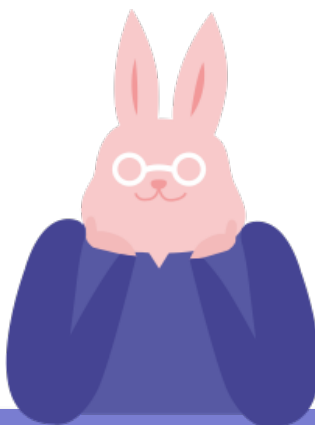
2016. 11. 22.

신현규



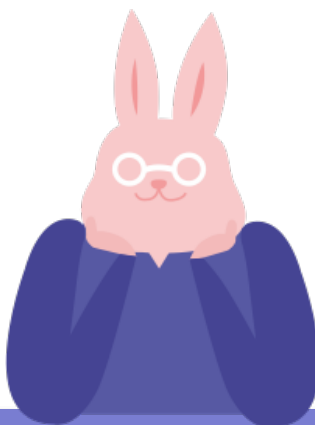
피드백

- 수업은 어느정도 이해 되고, 솔루션도 보면 어느정도 알겠는데, 문제는 못 풀겠어요
- 프로그래밍이 익숙하지 않아요 / 라이브러리 설명이 필요해요
- 문제가 너무 많아요. 다 풀어야 하나요?
- 문제 하나 푸는데 시간이 너무 오래걸려요
- 저 빼고 다 잘하는 것 같아요
- 수업시간을 좀 더 늘릴 수 없나요?
- Class가 갑자기 웬말이에요?
- ICPC 수업은 왜 하는거예요?



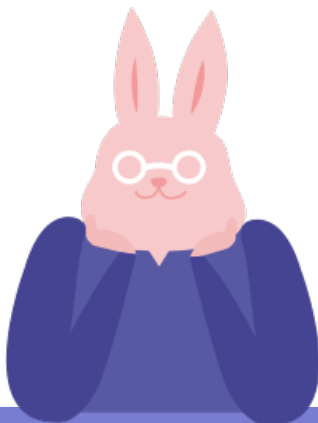
왜 ICPC인가?

- 우리의 목적은 Data science / Machine learning



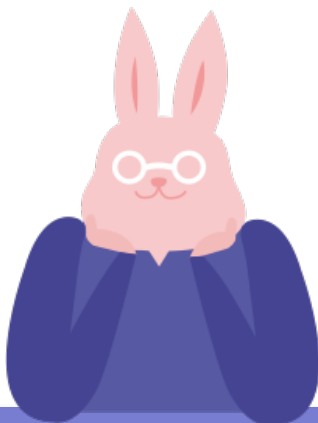
왜 ICPC인가?

- 우리의 목적은 Data science / Machine learning
- 현재 우리가 겪고 있는 어려움
 - 프로그래밍이 어렵다
 - 알고리즘이 어렵다
 - 알고리즘을 어떻게 코드로 옮겨야 하는지 모르겠다
 - 알고리즘은 알겠는데, 문제를 어떻게 풀어야 할지 모르겠다
 - 풀이까지 알겠는데 코딩을 못하겠다



왜 ICPC인가?

- 우리의 목적은 Data science / Machine learning
- 현재 우리가 겪고 있는 어려움
 - 프로그래밍이 어렵다
 - 데이터 분석 알고리즘이 어렵다
 - 데이터 분석 알고리즘을 어떻게 코드로 옮겨야 하는지 모르겠다
 - 데이터 분석 알고리즘은 알겠는데, 문제를 어떻게 풀어야 할지 모르겠다
 - 풀이까지 알겠는데 코딩을 못하겠다



Example (SK Code Sprint 14)

커피빈

전체 이미지 지도 뉴스 동영상 더보기 검색 도구

검색결과 약 893,000개 (0.58초)

관련검색: coffeebean coupon coffeebean korea coffee bean logo coffee bean promo code coffee bean sign

The Coffee Bean & Tea Leaf Official Store
www.coffeebean.com/ 이 페이지 번역하기
 Born and brewed in Southern California since 1963, The Coffee Bean & Tea Leaf® delivers the taste and aroma of the world's best coffees and teas.
[Locator](#) · [Beverage menu](#) · [Careers](#) · [Coffee](#)

커피빈 앤드 티리프 회사

커피빈 앤드 티리프 또는 커피빈은 미국 로스앤젤레스에 본사가 있는 인터내셔널 커피 앤드 티 유한책임회사에서 운영하고 있는 세계에서 가장 오래된 커피 및 홍차 전문점 프랜차이즈 중의 하나이다. 최초의 마당은 1963년에 로스앤젤레스의 브랜트우드에서 허버트 B. 하이만과 모나 하이만이 개점하였다. 현재 미국 남서부와 하와이 주까지 아우르는 점포망을 갖고 있다. [위키백과](#)

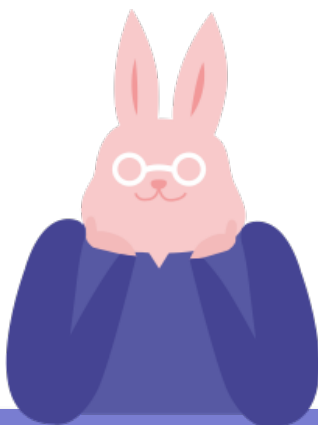
CEO: 존 로슨 (2014년 1월 1일~)
 창립자: 허버트 하이만
 창립: 1963년, 미국 캘리포니아 주 로스앤젤레스 브랜트우드
 주소: 미국 캘리포니아 주 로스앤젤레스

A 커피빈코리아 대전카이스트점
 대학로 291 · 042-867-3751
 오후 9:30까지

https://codesprint.skplanet.com/2014/participation/round1_intro

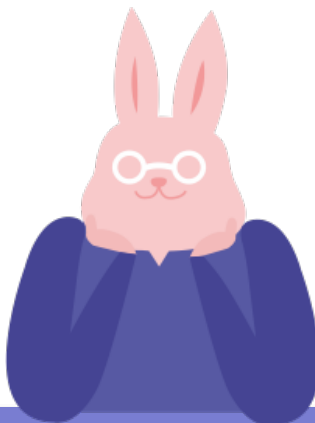
Example (SK Code Sprint 14)

- 검색어가 주어졌을 때, 사용자의 의도대로 정확히 잘라라
 - “coffeebean” → “coffee bean”
- Training set 으로 컴퓨터를 학습시켜 test set의 정답을 최대한 많이 맞히는 사람이 승리
 - Training set은 10만 문장, test set은 1만개의 string
 - Training 10분, testing 10초



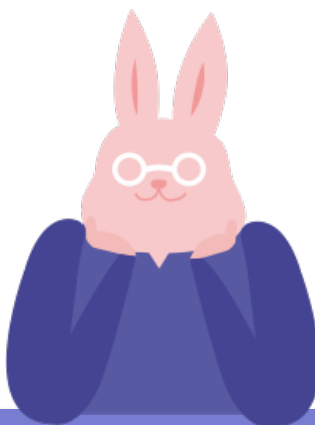
문제 풀이 순서

1. 문제를 이해한다
2. 문제를 풀기 위한 알고리즘을 디자인 한다
3. 알고리즘이 문제를 정확히 푼다는 것을 증명한다
4. 알고리즘이 제한 시간 내에 정답을 낸다는 것을 증명한다
5. 정확하게 구현한다



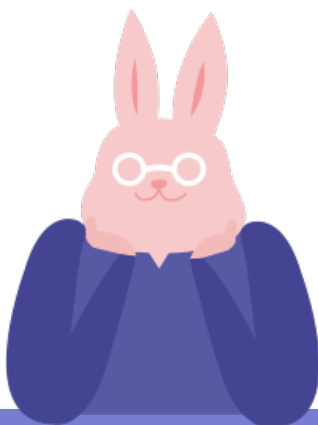
문제 풀이 순서

1. 문제를 이해한다
2. 문제를 풀기 위한 알고리즘을 디자인 한다 : 굉장히 어렵다
3. 알고리즘이 문제를 정확히 푼다는 것을 증명한다 : 증명할 수 없음
4. 알고리즘이 제한 시간 내에 정답을 낸다는 것을 증명한다 : 가능한 함
5. 정확하게 구현한다



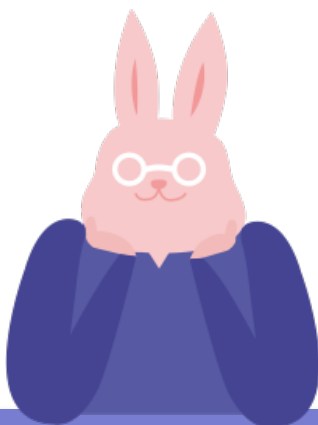
Machine Learning 을 배울 준비

- 알고리즘 디자인 Training 을 위해서
 - 이미 디자인 되어있는 여러 알고리즘을 배우고,
 - 이를 직접 구현해보자



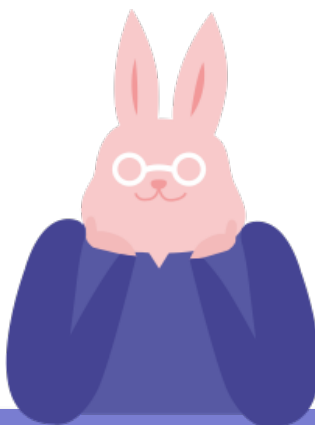
Machine Learning 을 배울 준비

- 알고리즘 디자인 Training 을 위해서
 - 이미 디자인 되어있는 여러 알고리즘을 배우고,
 - 이를 직접 구현해보자
- 정확성을 증명할 수 없는 문제를 풀기 위해서
 - 일단 정답이 존재하는 문제부터 제대로 증명하고 풀어보자



Machine Learning 을 배울 준비

- 알고리즘 디자인 Training 을 위해서
 - 이미 디자인 되어있는 여러 알고리즘을 배우고,
 - 이를 직접 구현해보자
- 정확성을 증명할 수 없는 문제를 풀기 위해서
 - 일단 정답이 존재하는 문제부터 제대로 증명하고 풀어보자
- 제한시간 내에 정답을 낸다는 것을 보이기 위해서
 - 간단한 알고리즘의 시간복잡도부터 계산해보자



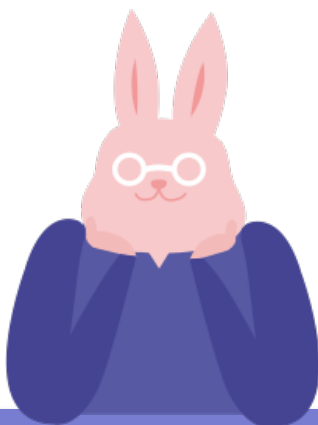
어려워야 정상

- 쉽지 않음. 하지만 꼭 필요한 과정이라고 생각함
- 다들 너무 열심히 듣고, 열심히 따라와 주셔서 감사합니다 :)



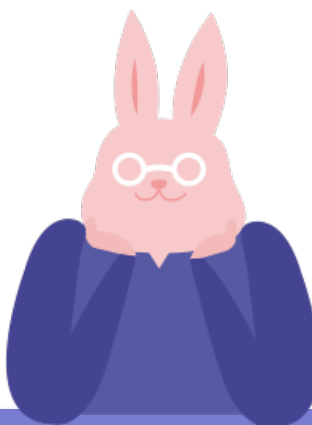
요약

- 답이 존재하는 문제를 먼저 풀어보고, 머신러닝 배울 준비 하자
- 6개월 뒤에는 다른 사람이 되어 있다는 것을 보장합니다



지난 시간 요약

- 알고리즘의 정확성 증명
- 시간복잡도
- 자료구조
 - 스택
 - 큐
 - 트리
 - 힙



[활동문제 0] 달팽이 숫자 출력

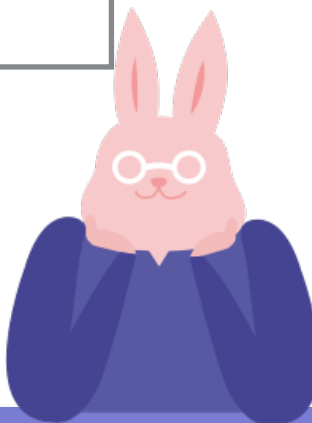
- 아래와 같이 달팽이 모양으로 숫자를 출력

입력의 예

5

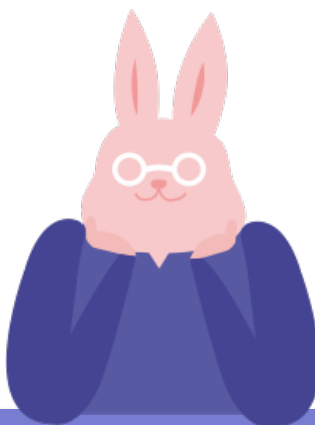
출력의 예

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



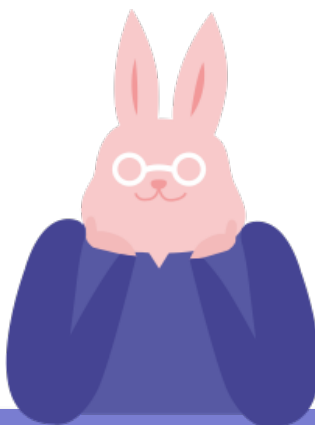
커리큘럼

1. 재귀호출, 추상화
2. 시간복잡도, 알고리즘 정확성 증명, 자료구조
3. 분할정복법, 탐욕적 기법
4. 동적계획법 1
5. 동적계획법 2
6. 그래프 이론 1
7. 그래프 이론 2
8. 세계 여러 기업의 입사 인터뷰 문제 도전 (+ NP-Complete)



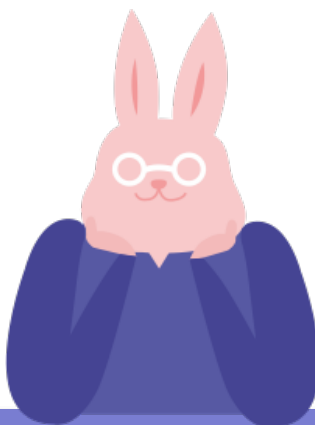
커리큘럼

1. 재귀호출, 추상화
2. 시간복잡도, 알고리즘 정확성 증명, 자료구조
3. 분할정복법, 탐욕적 기법
4. 동적계획법 1
5. 동적계획법 2
6. 그래프 이론 1
7. 그래프 이론 2
8. 세계 여러 기업의 입사 인터뷰 문제 도전 (+ NP-Complete)



분할정복법

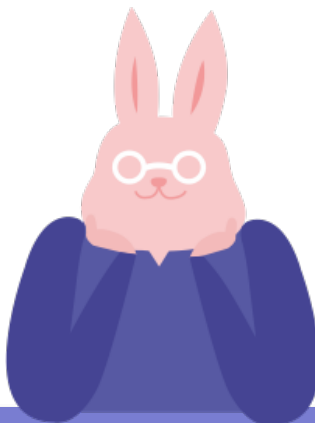
- 분할된 소문제를 해결한 후, 이를 합침으로써 전체 문제를 해결
- Recursion!!!!!!



Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---



Merge sort

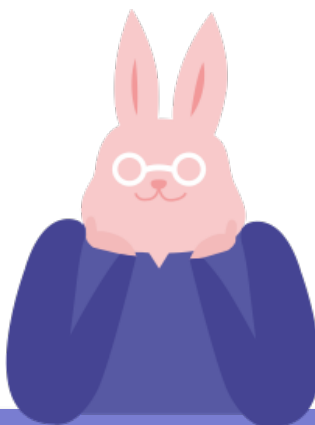
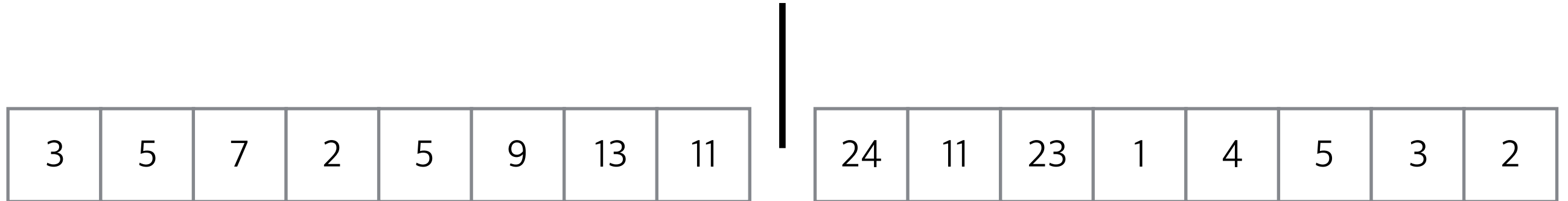
- $O(n \log n)$ 정렬의 가장 대표적인 예

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---



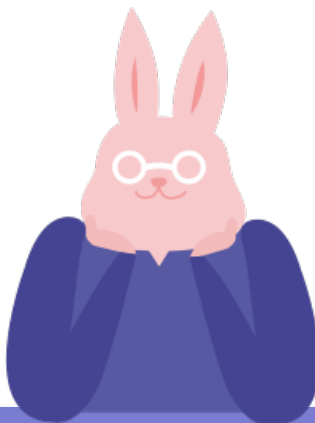
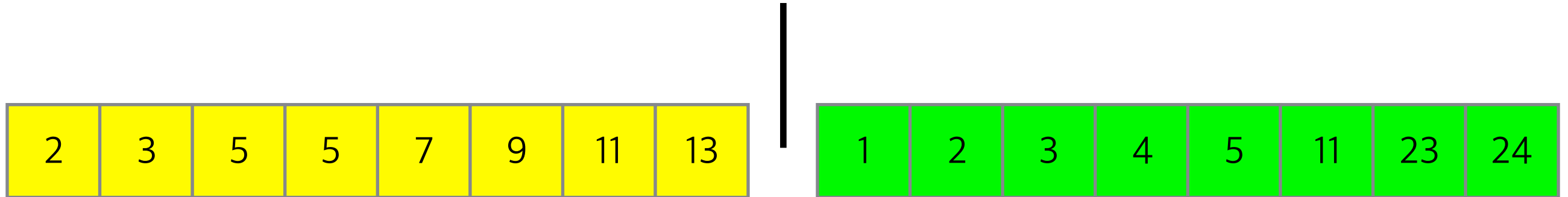
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



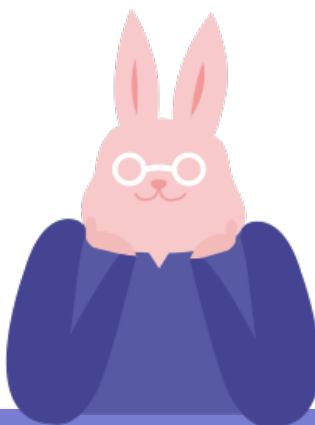
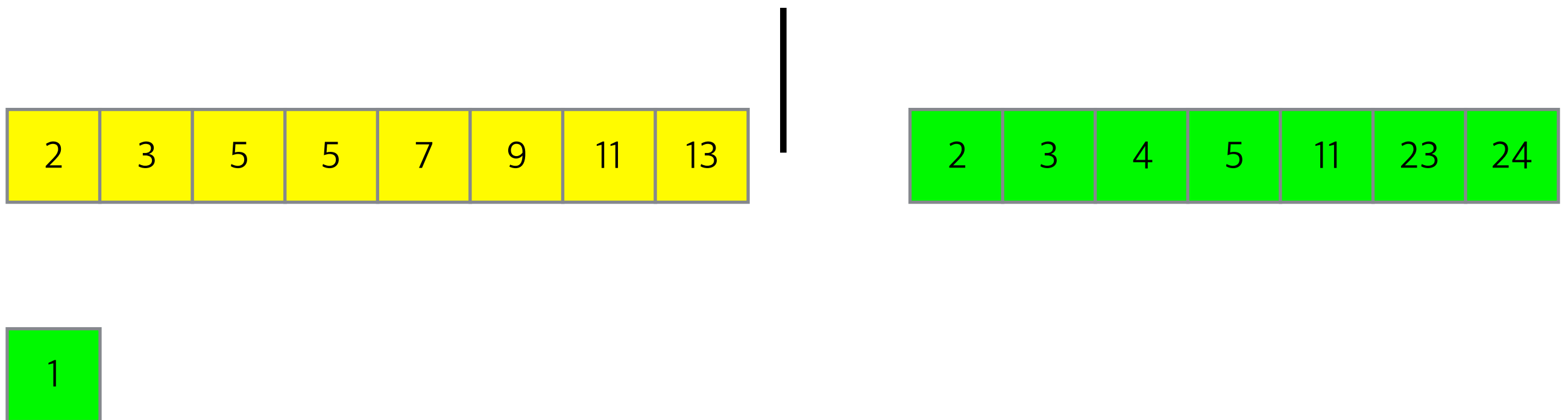
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



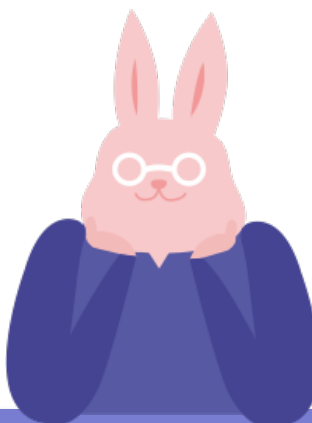
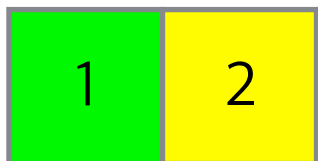
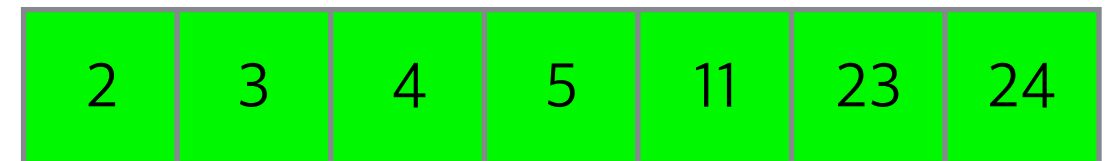
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



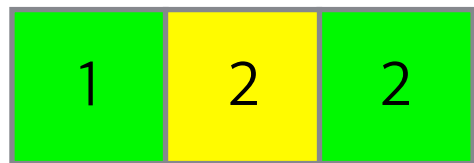
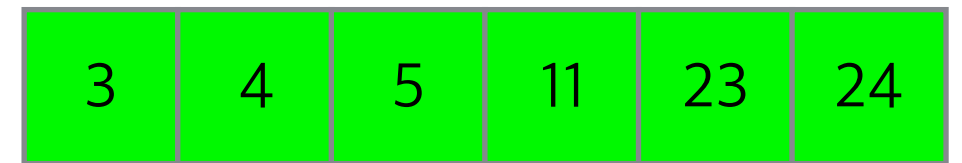
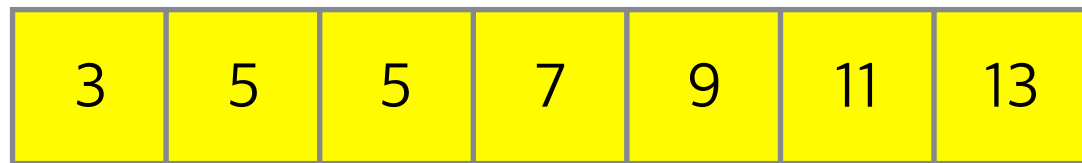
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



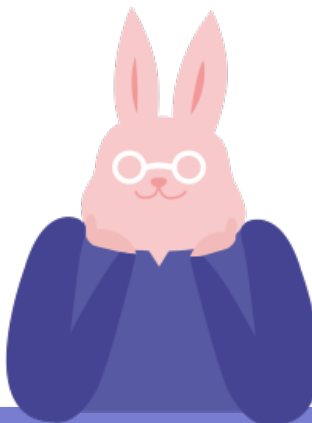
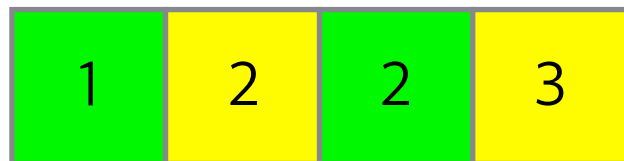
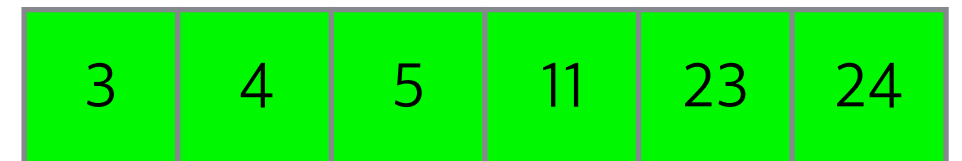
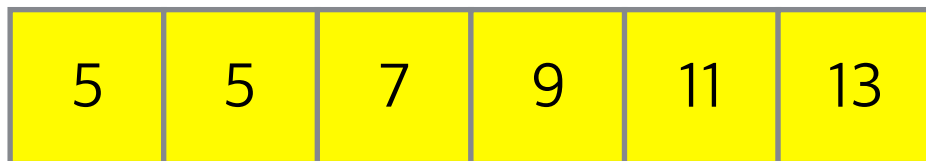
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



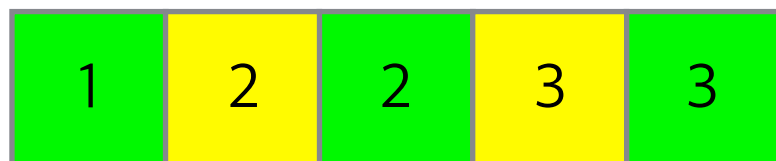
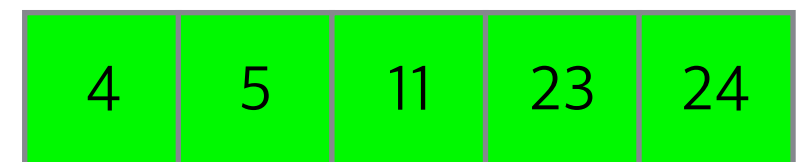
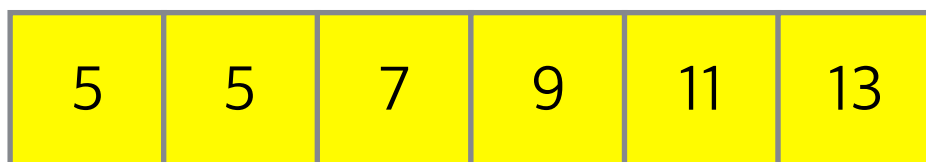
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



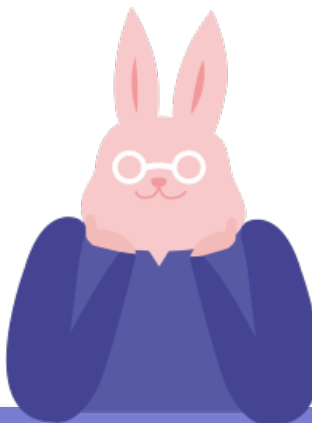
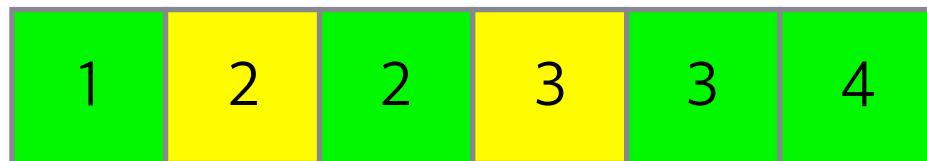
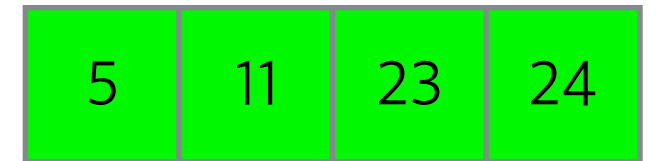
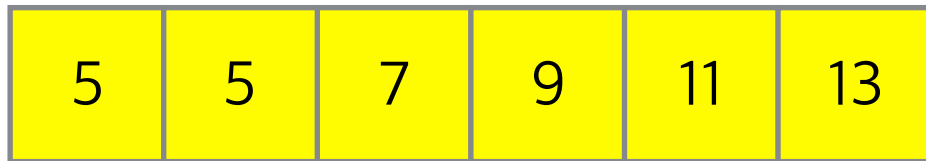
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



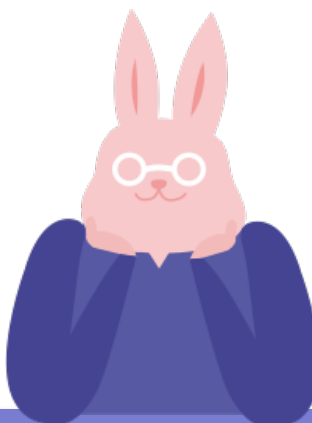
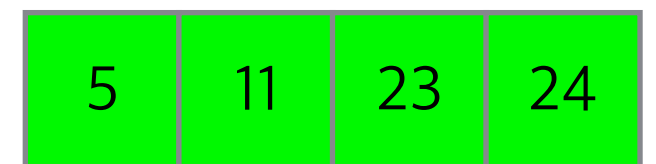
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



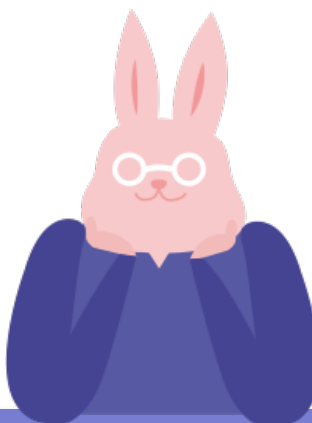
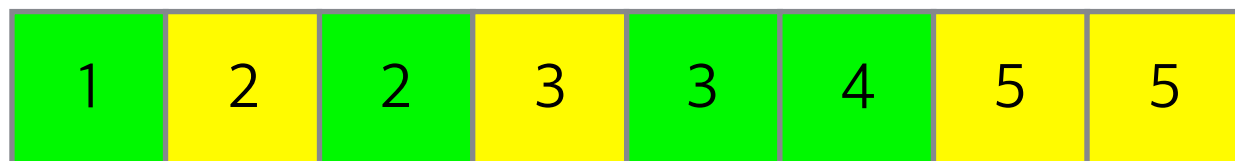
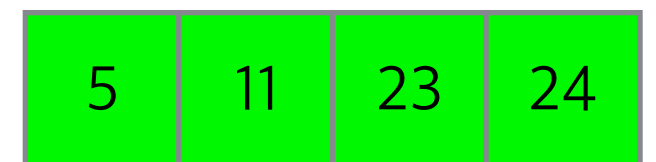
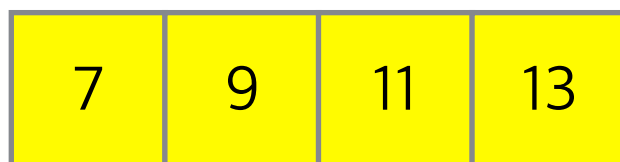
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



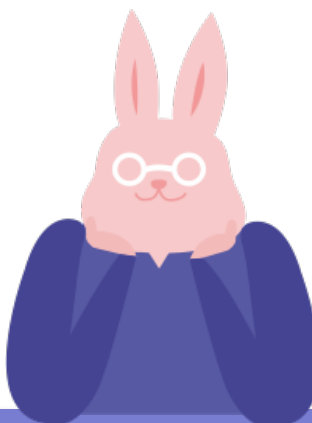
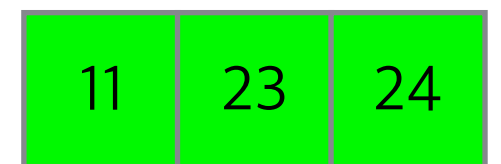
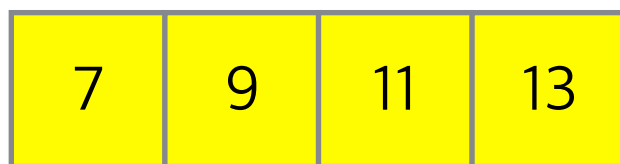
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



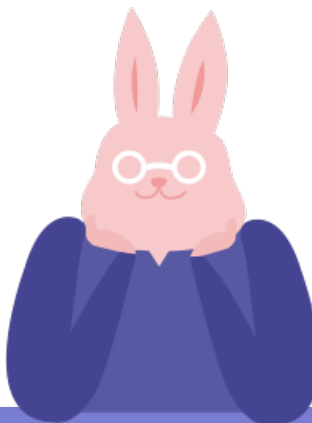
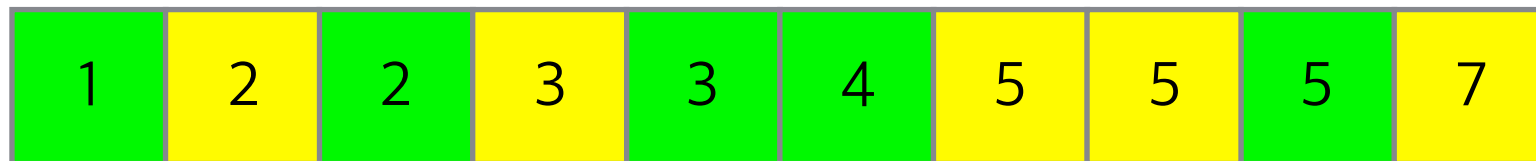
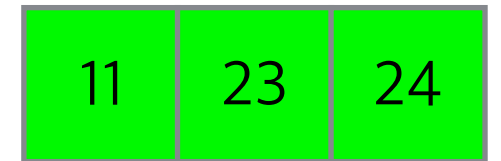
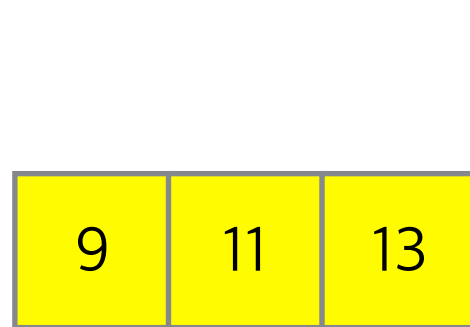
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



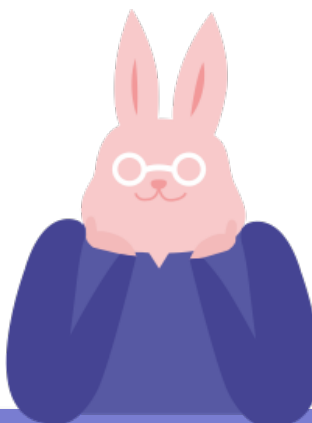
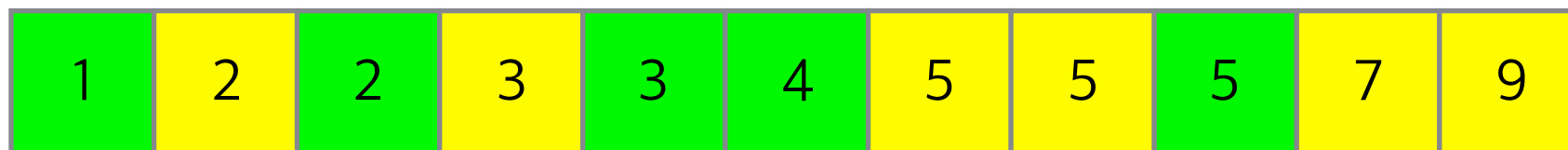
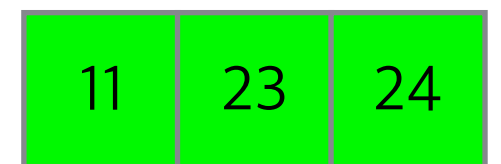
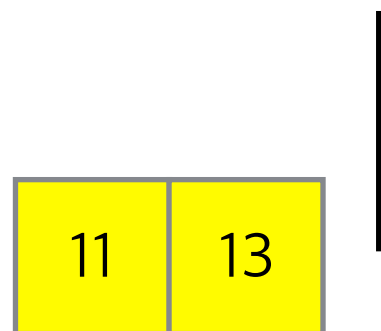
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



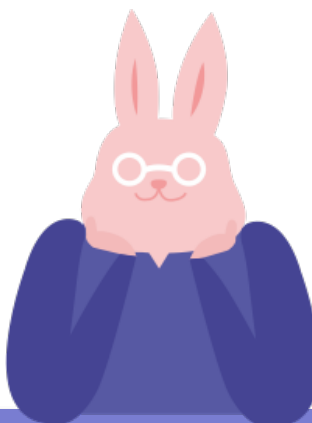
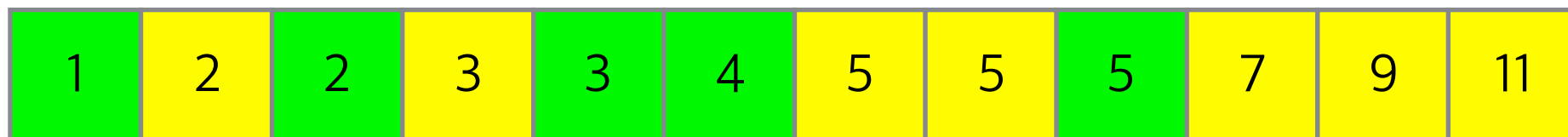
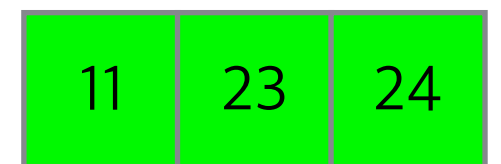
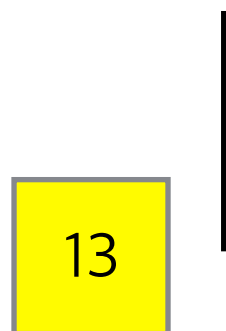
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



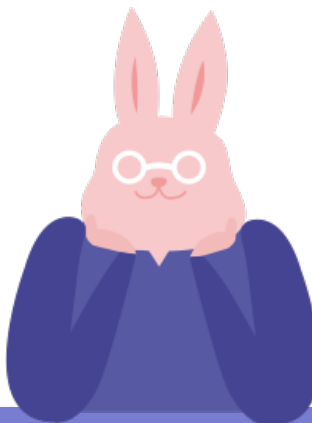
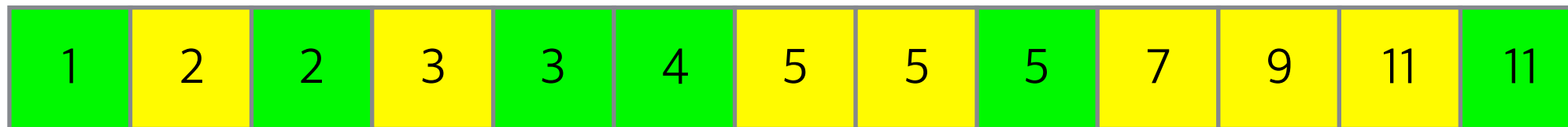
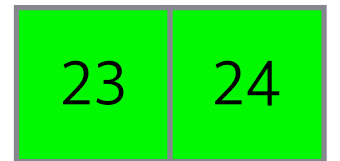
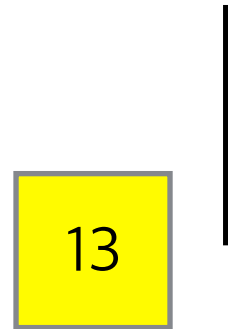
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예

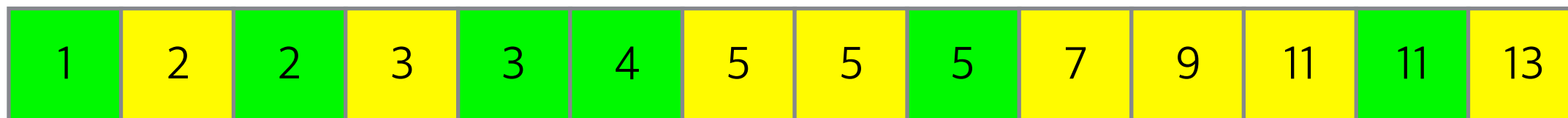


Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예

|

23	24
----	----



Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예

|

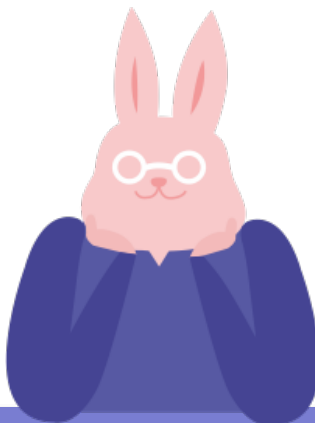
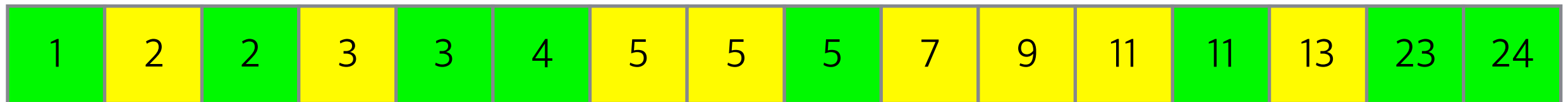
24



Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예

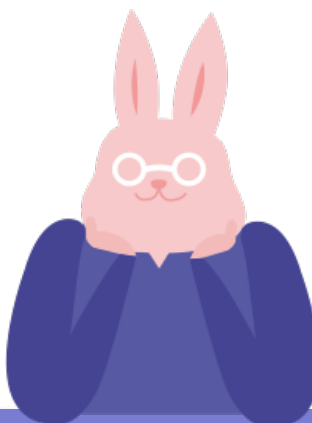
|



Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예

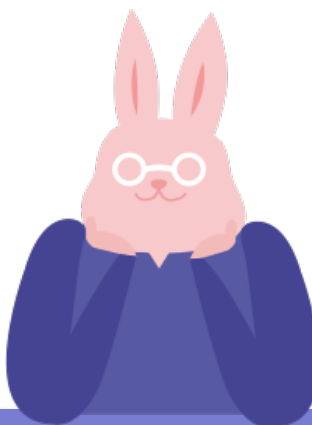
1	2	2	3	3	4	5	5	5	7	9	11	11	13	23	24
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----



Merge sort

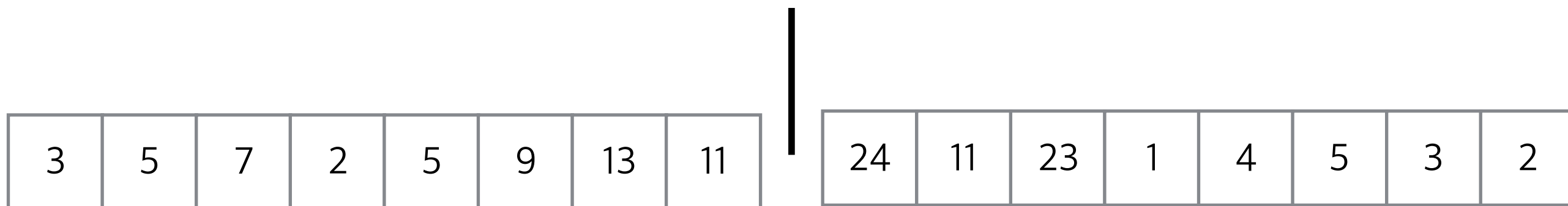
- $O(n \log n)$ 정렬의 가장 대표적인 예

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---



Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



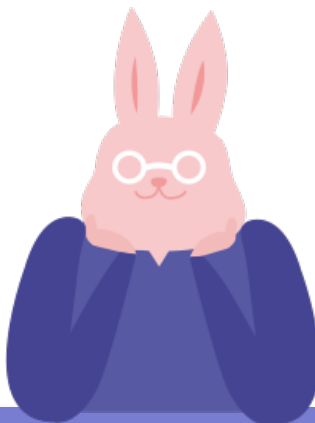
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예

3	5	7	2	5	9	13	11
---	---	---	---	---	---	----	----

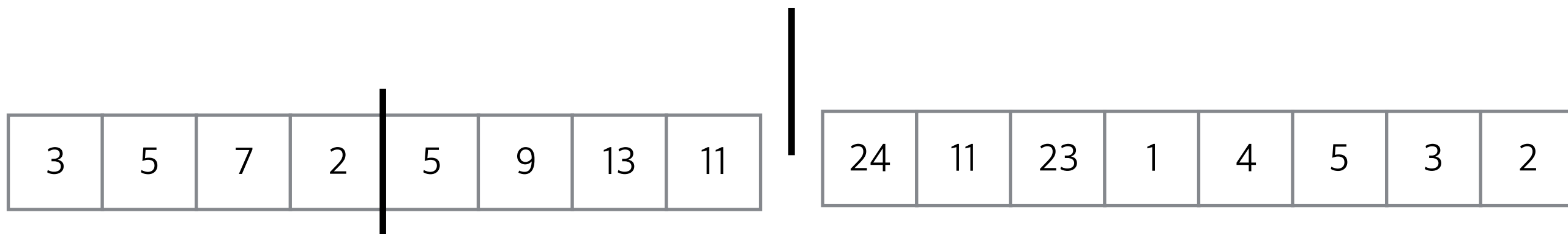
24	11	23	1	4	5	3	2
----	----	----	---	---	---	---	---

각각을 Merge sort로 정렬!



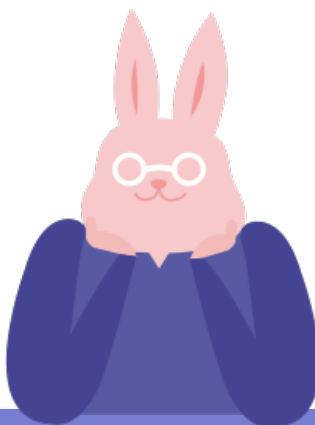
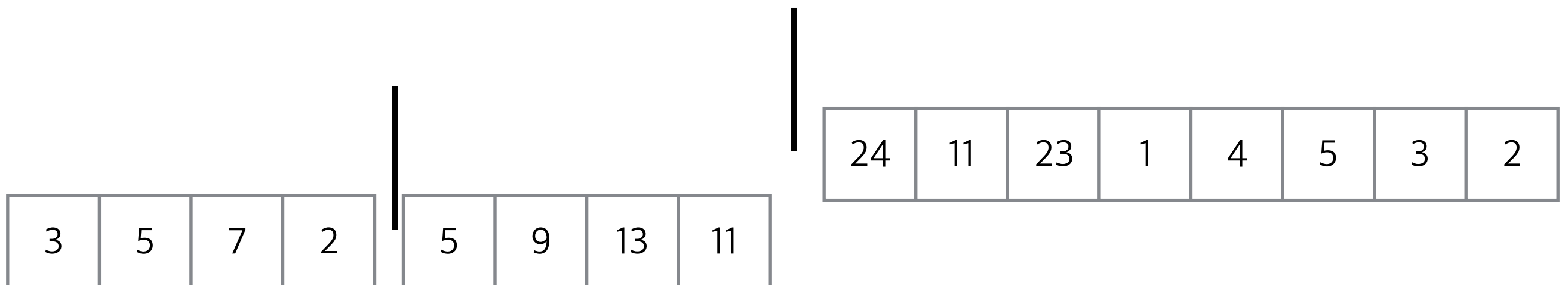
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



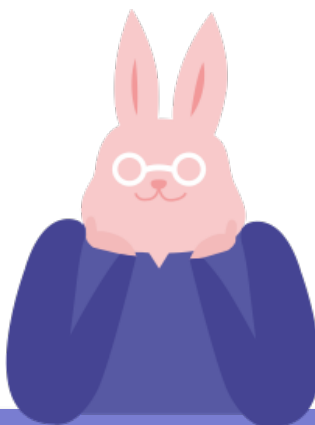
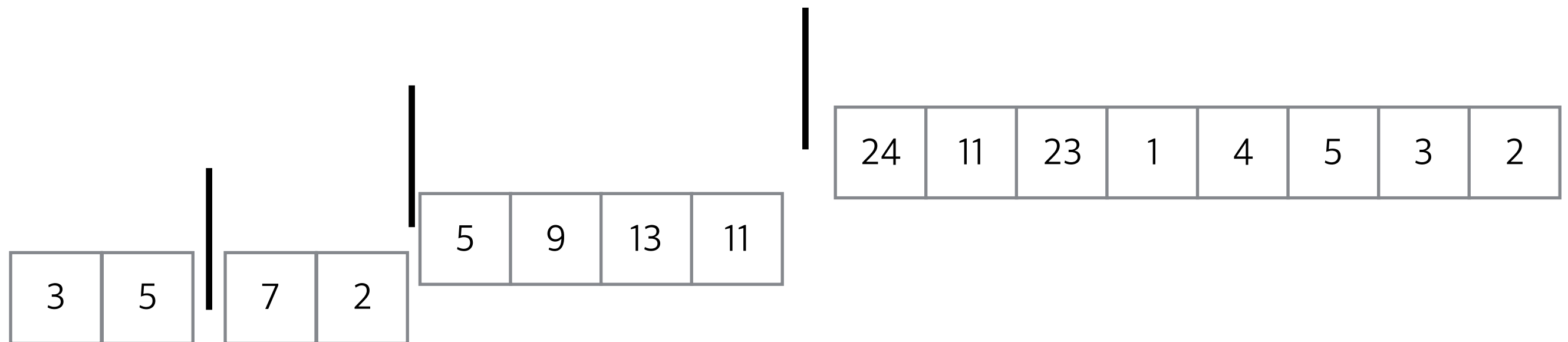
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



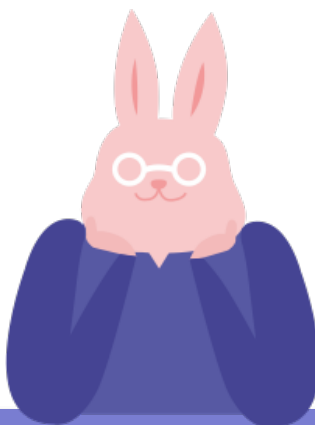
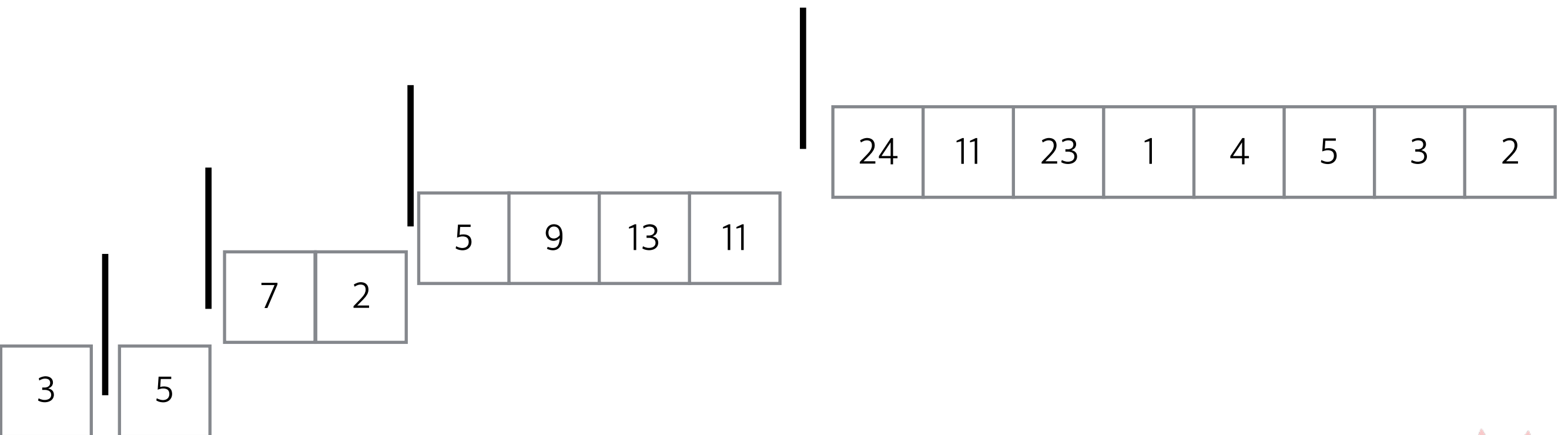
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



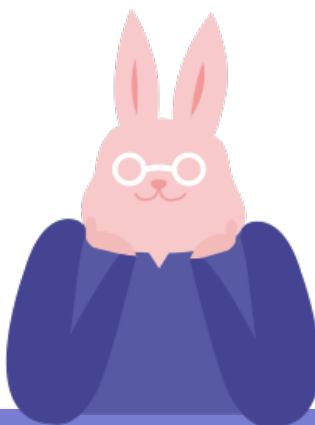
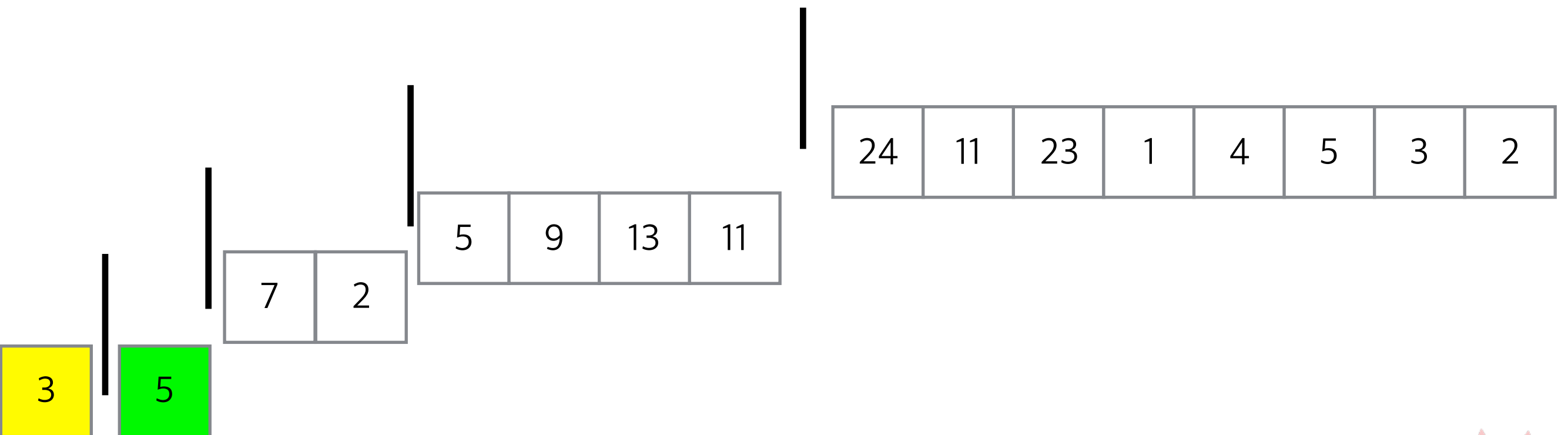
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



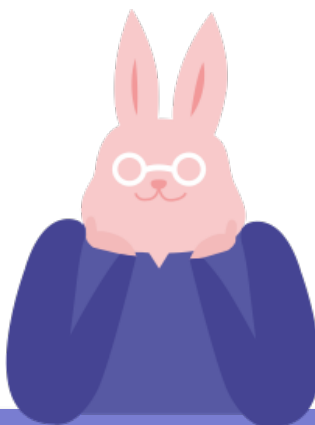
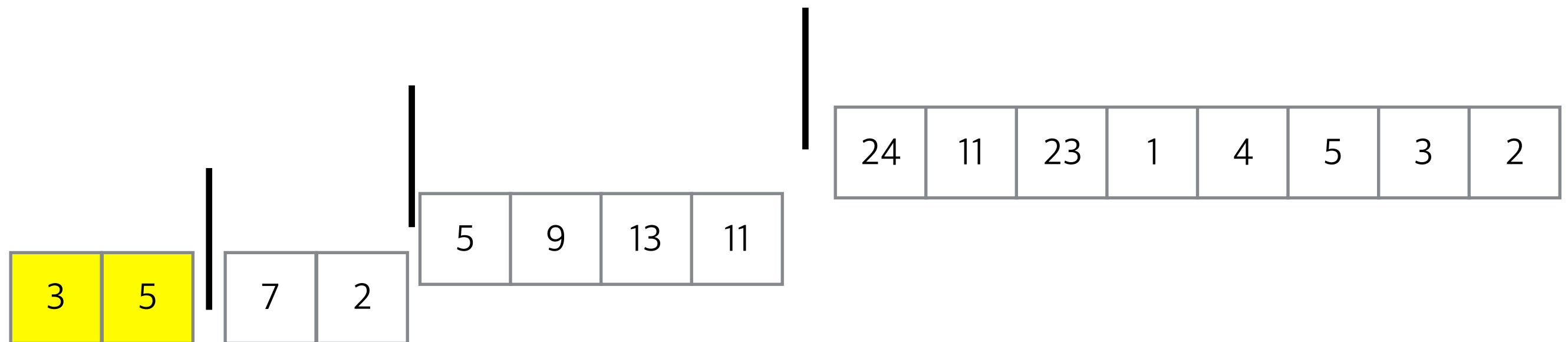
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



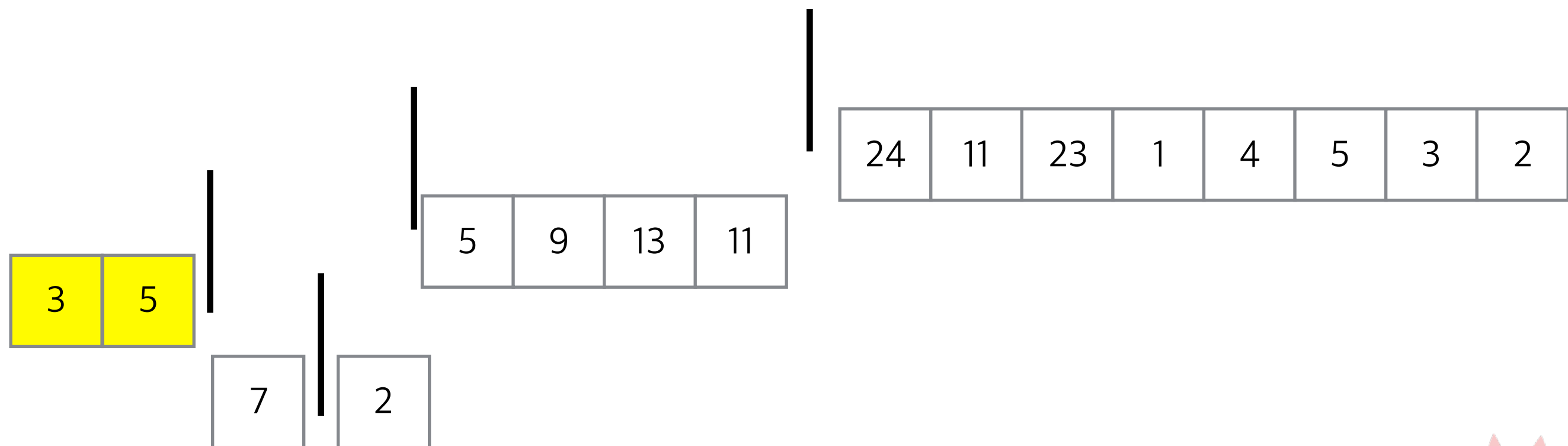
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



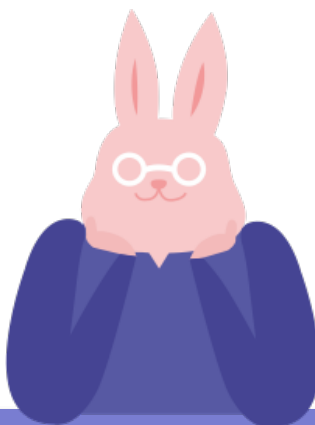
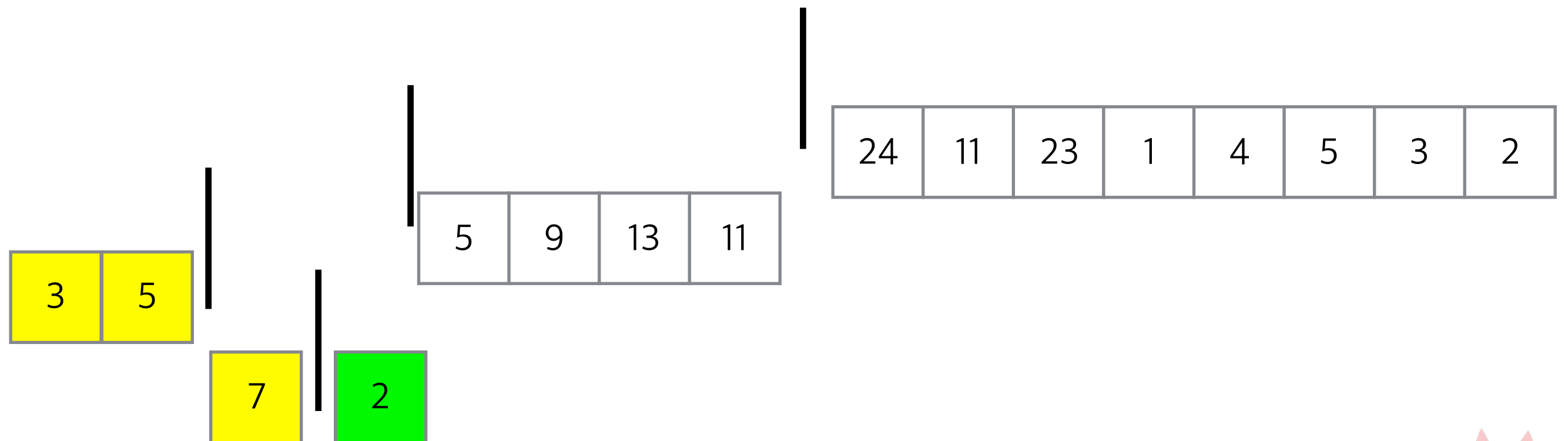
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



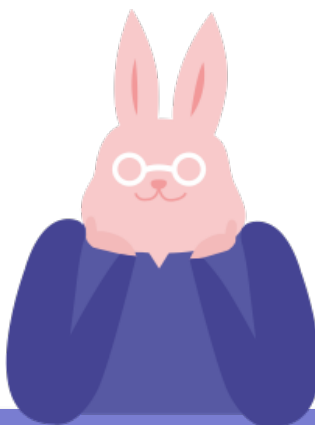
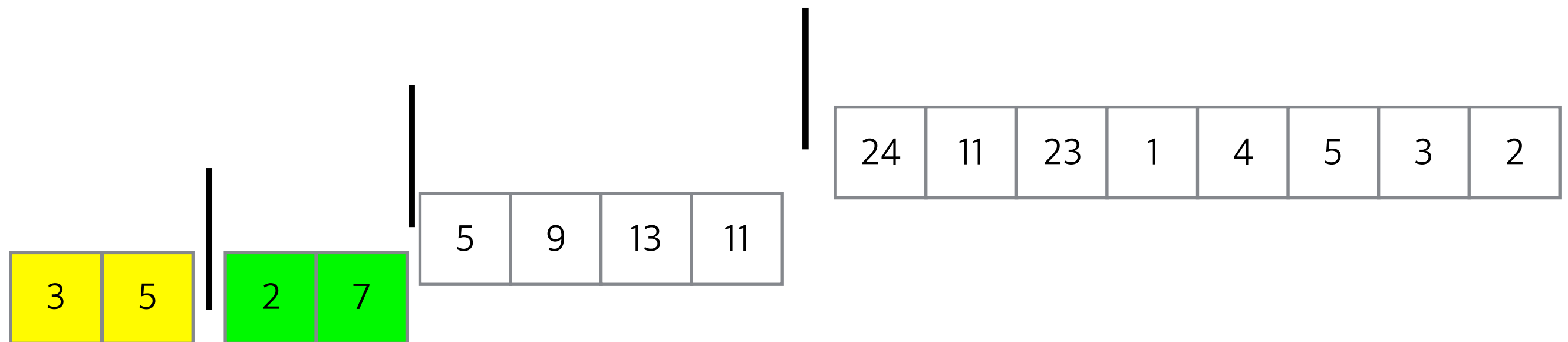
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



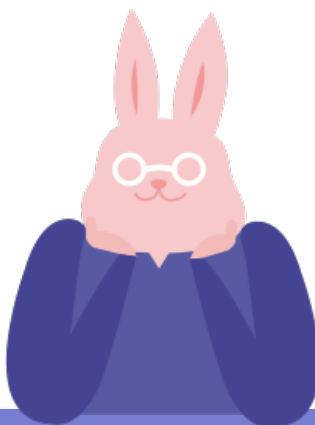
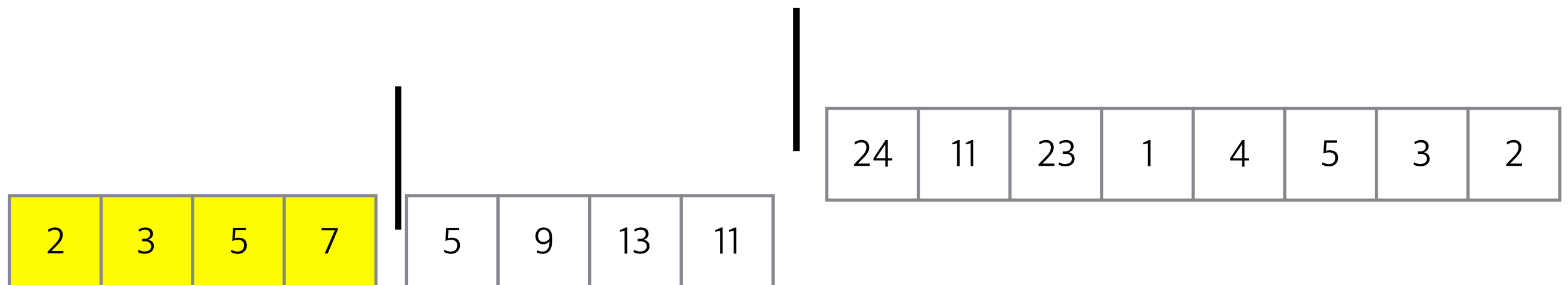
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



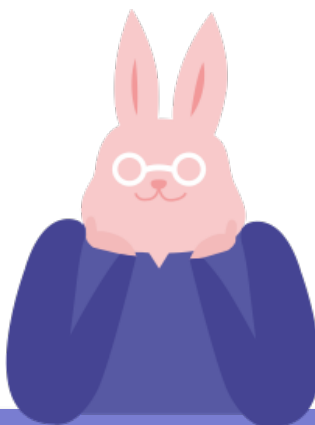
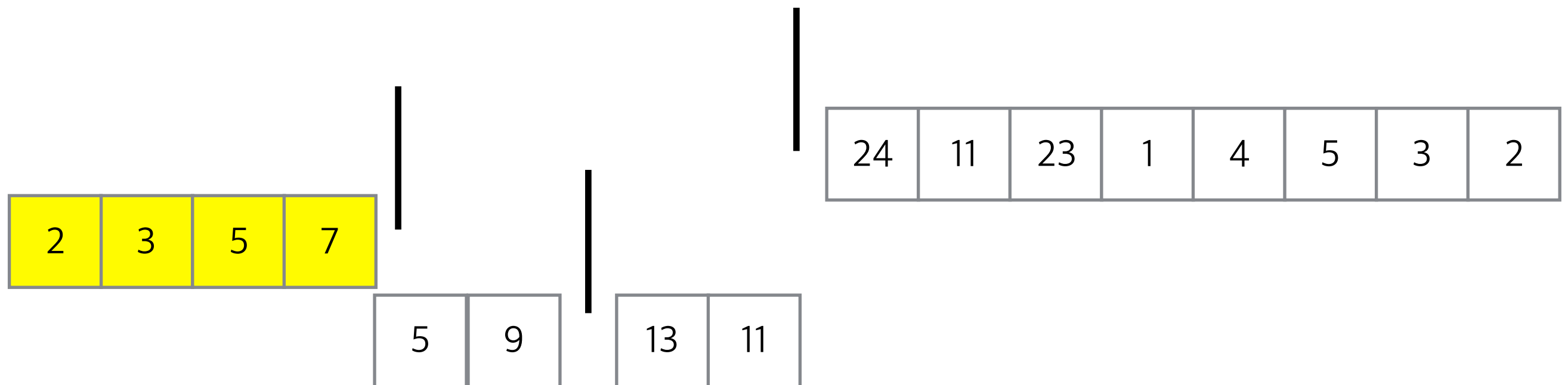
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



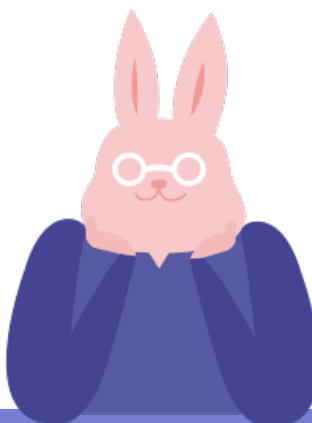
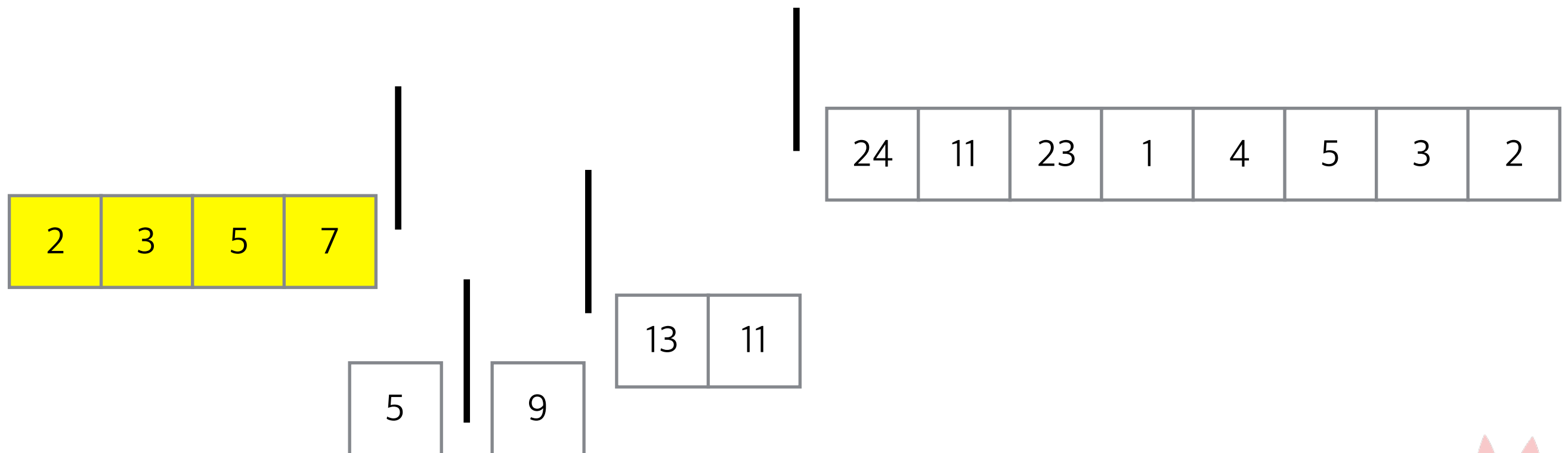
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



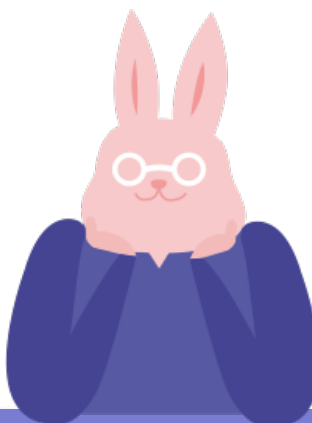
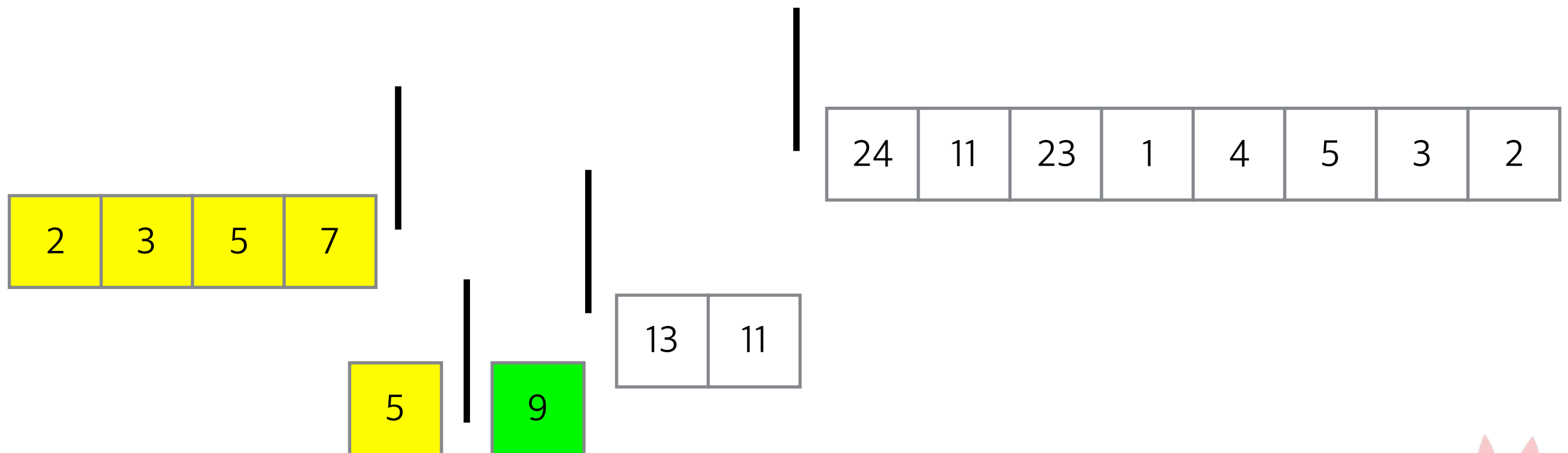
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



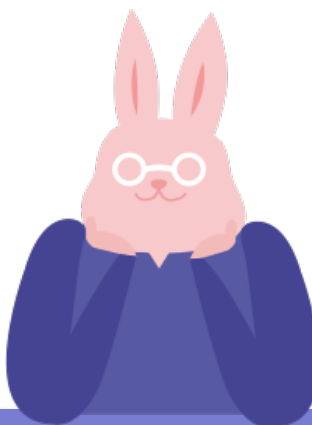
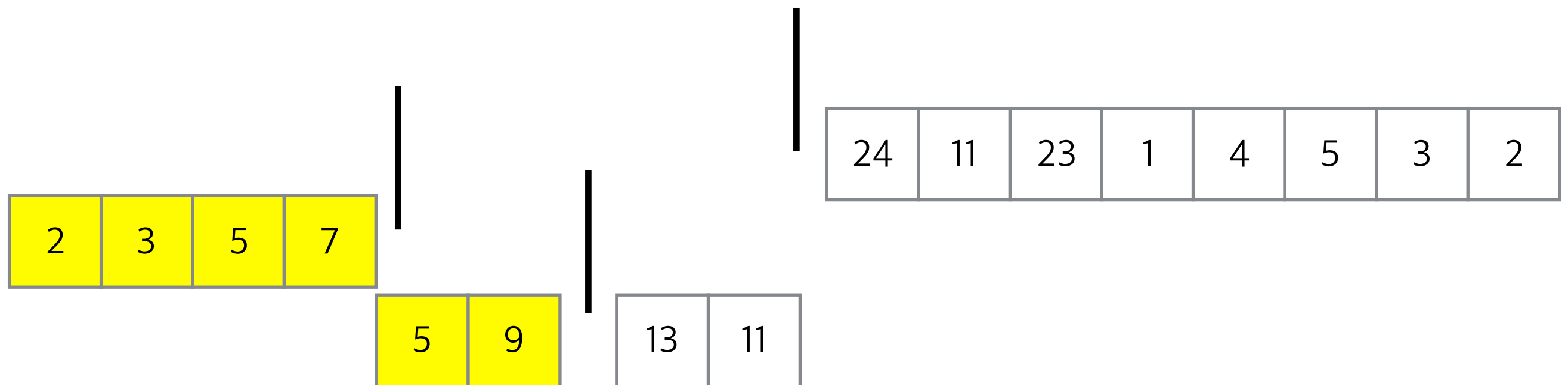
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



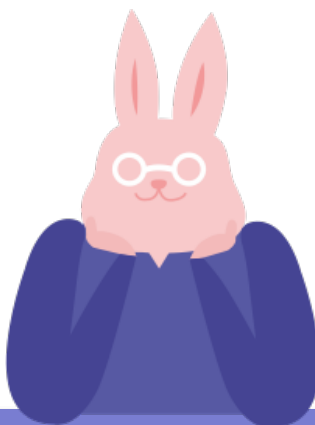
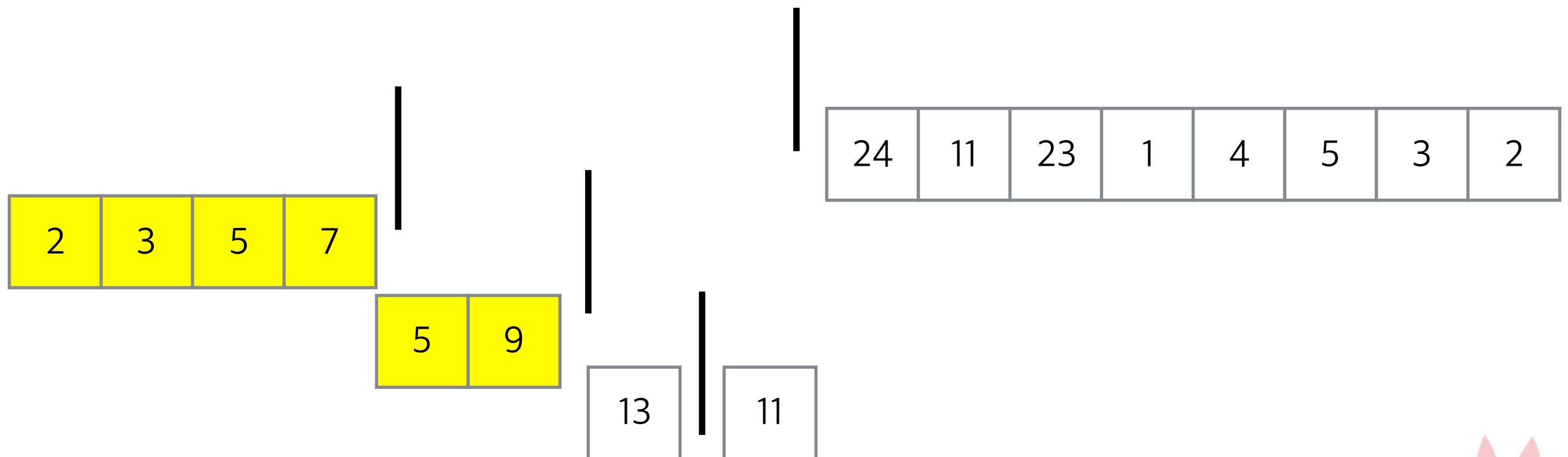
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



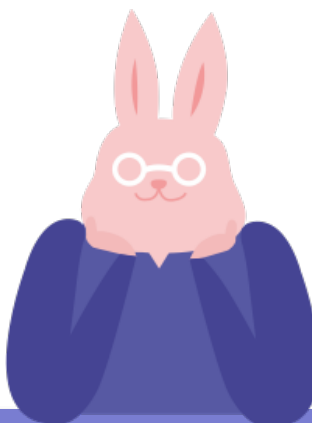
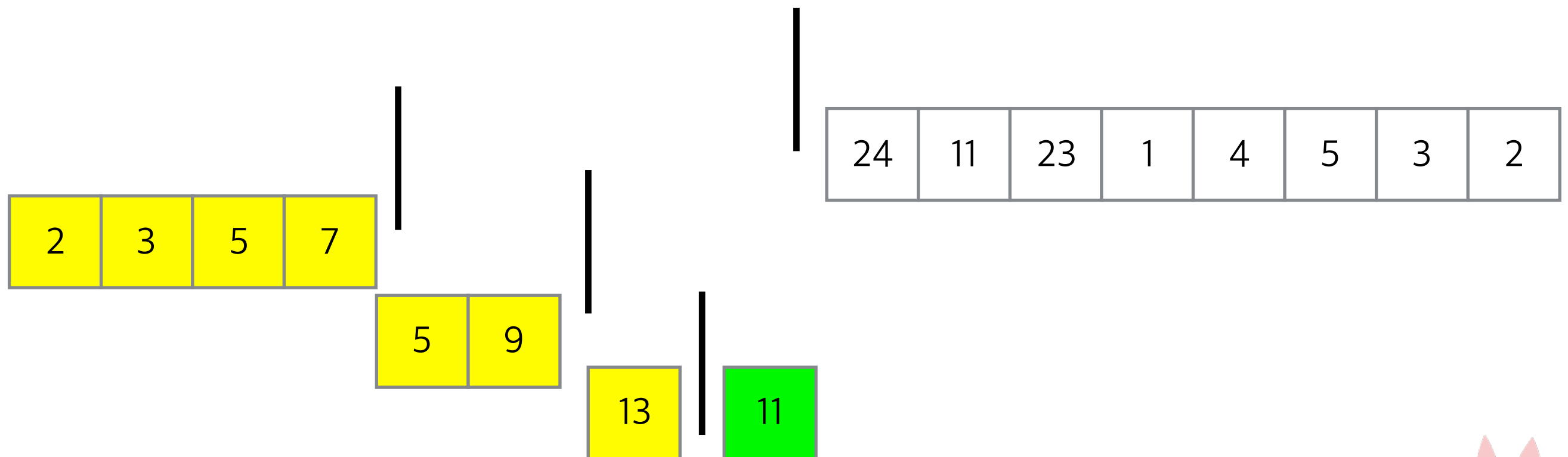
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



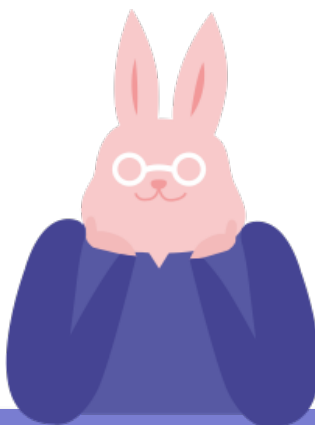
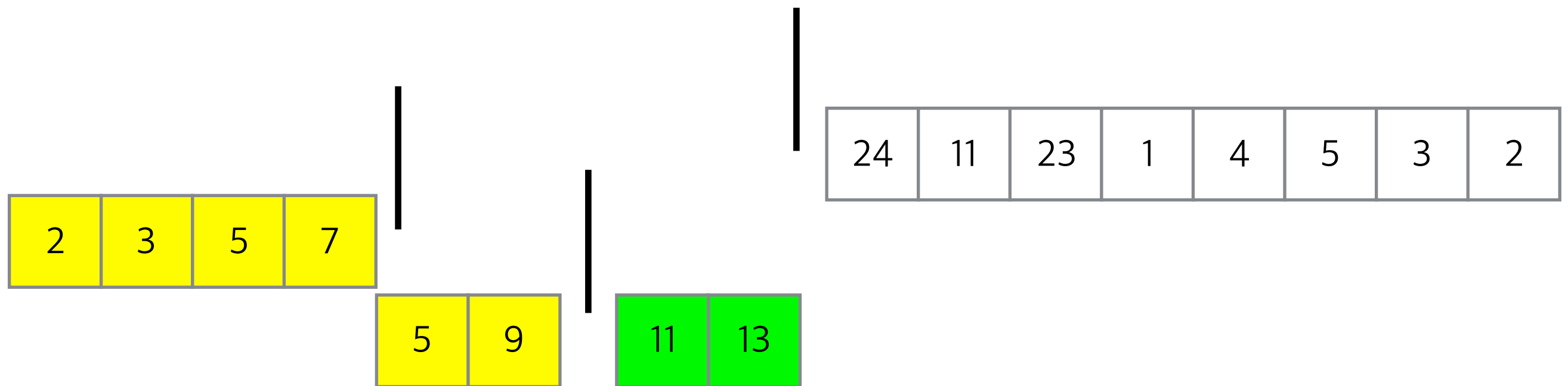
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



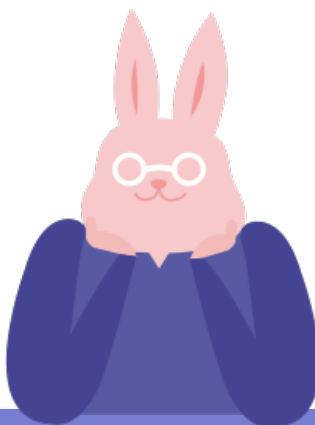
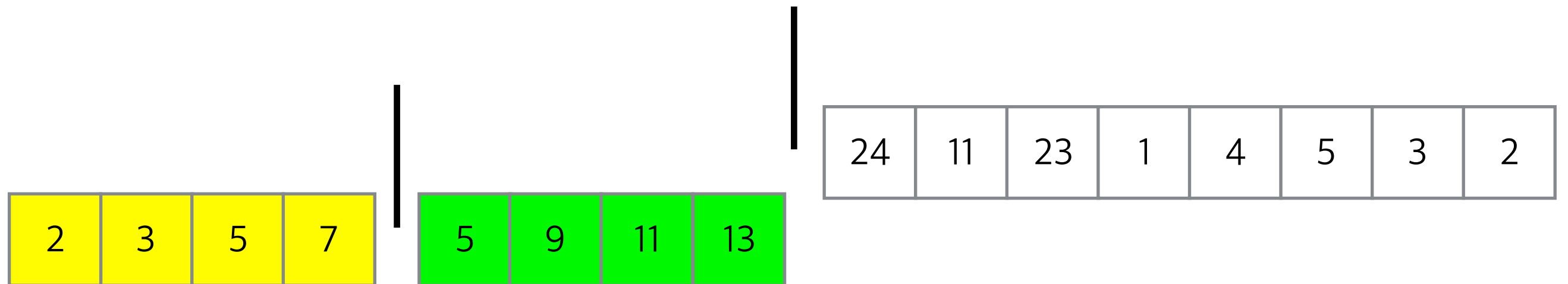
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



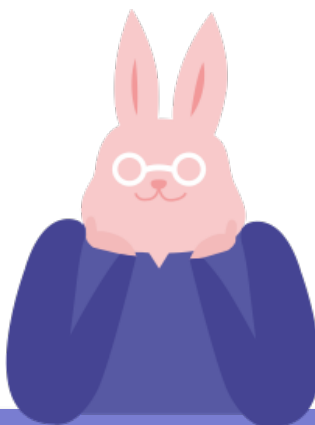
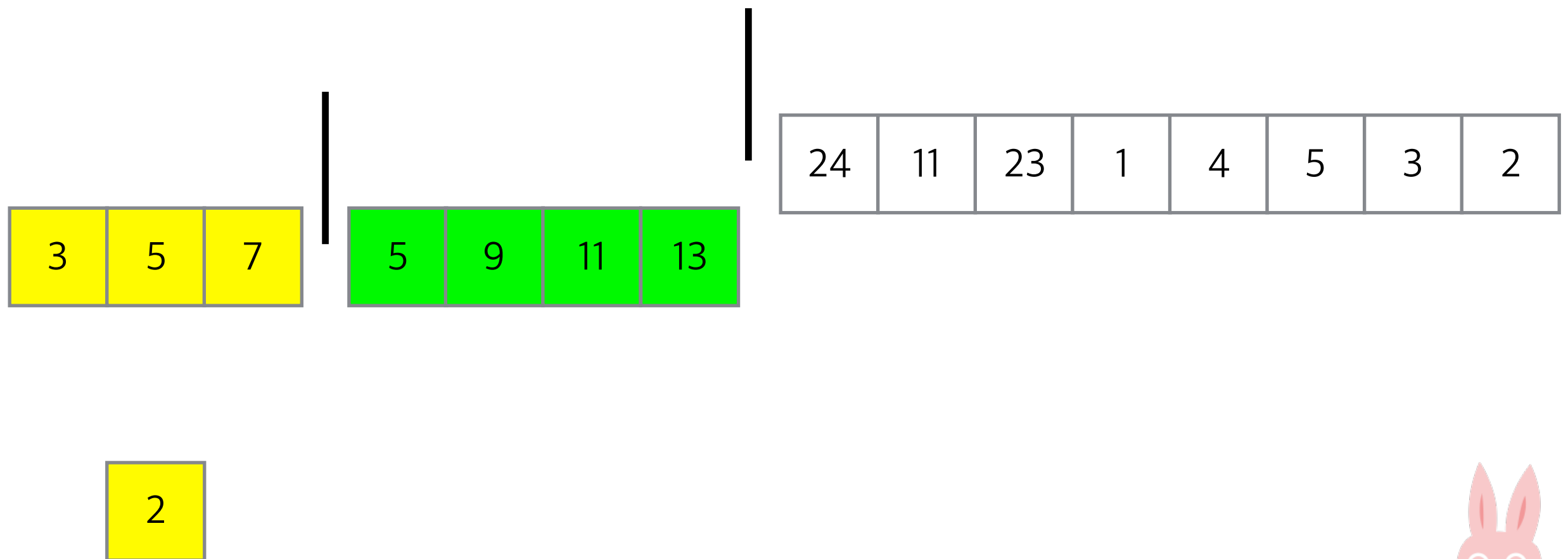
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



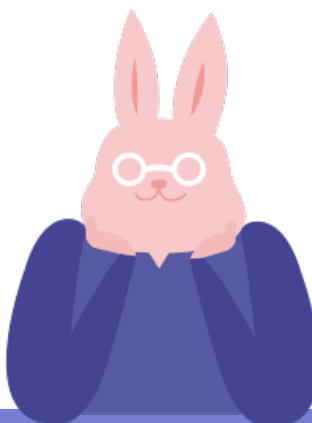
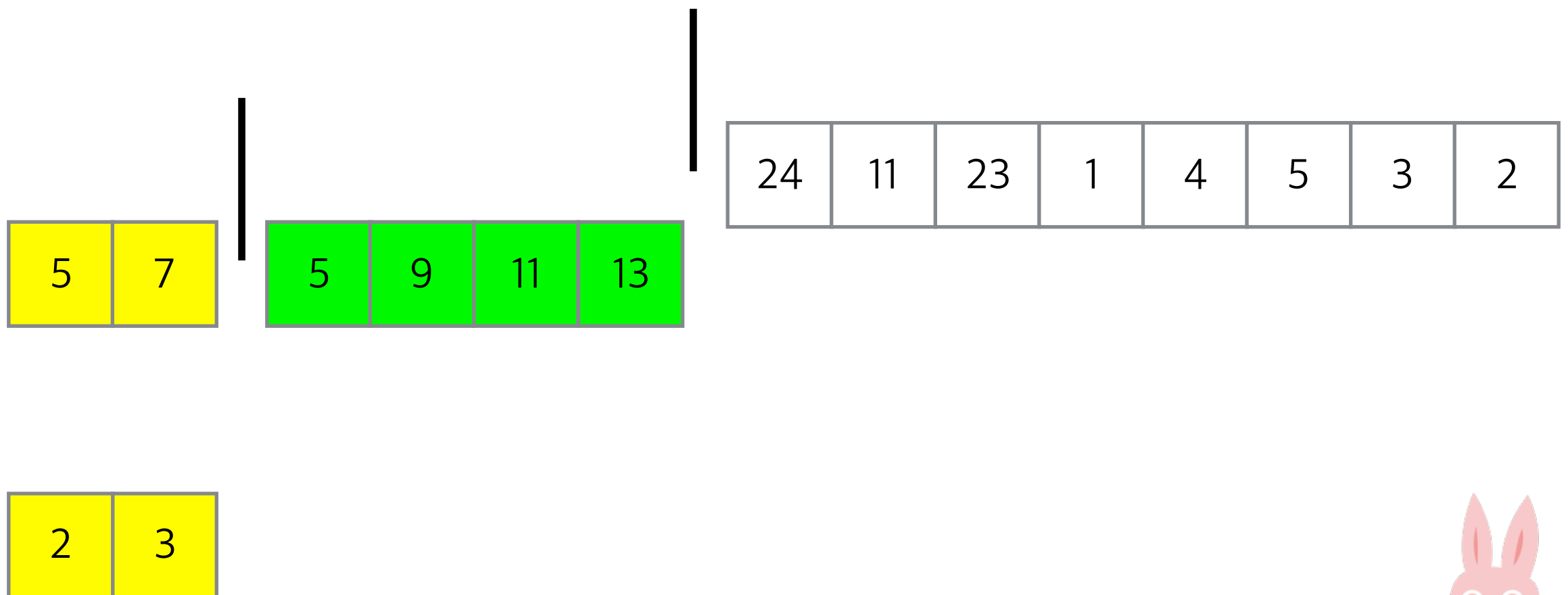
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



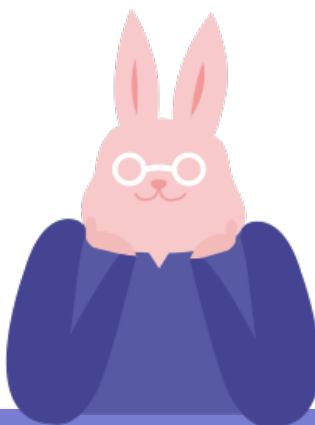
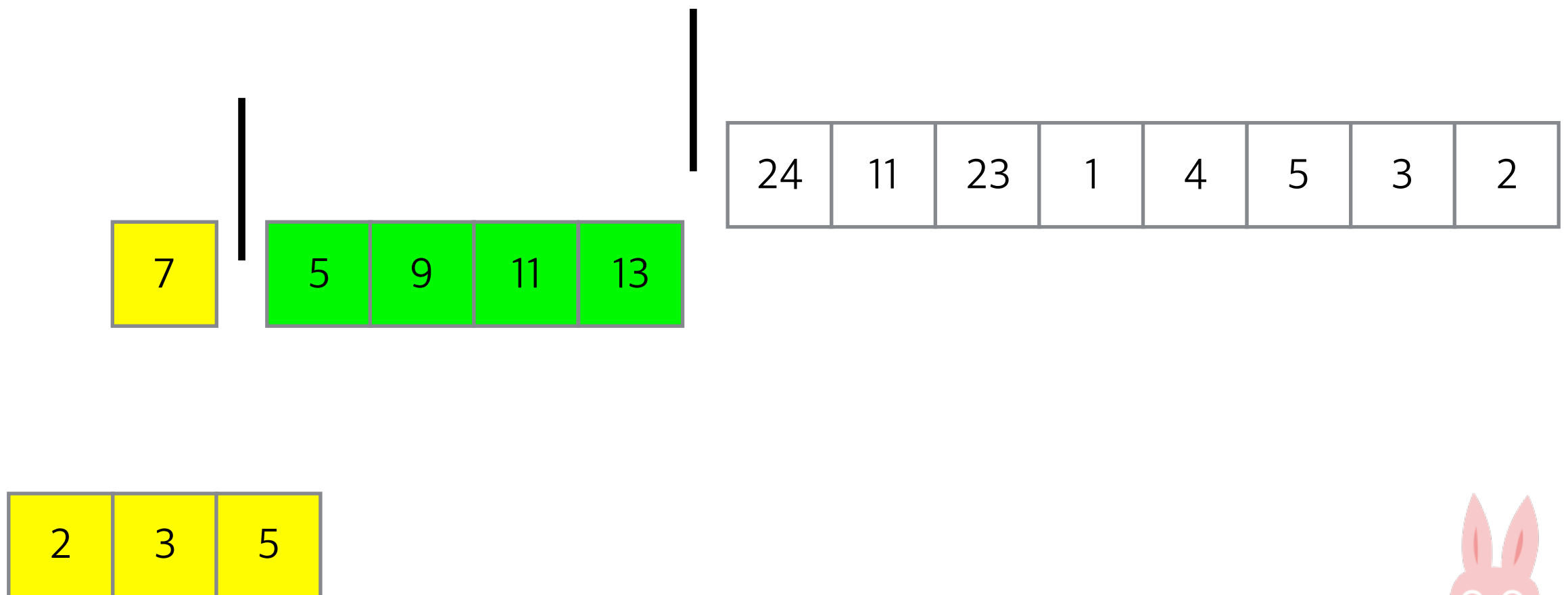
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



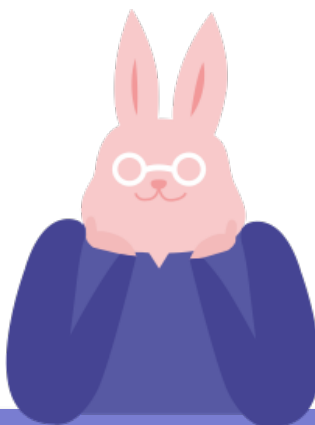
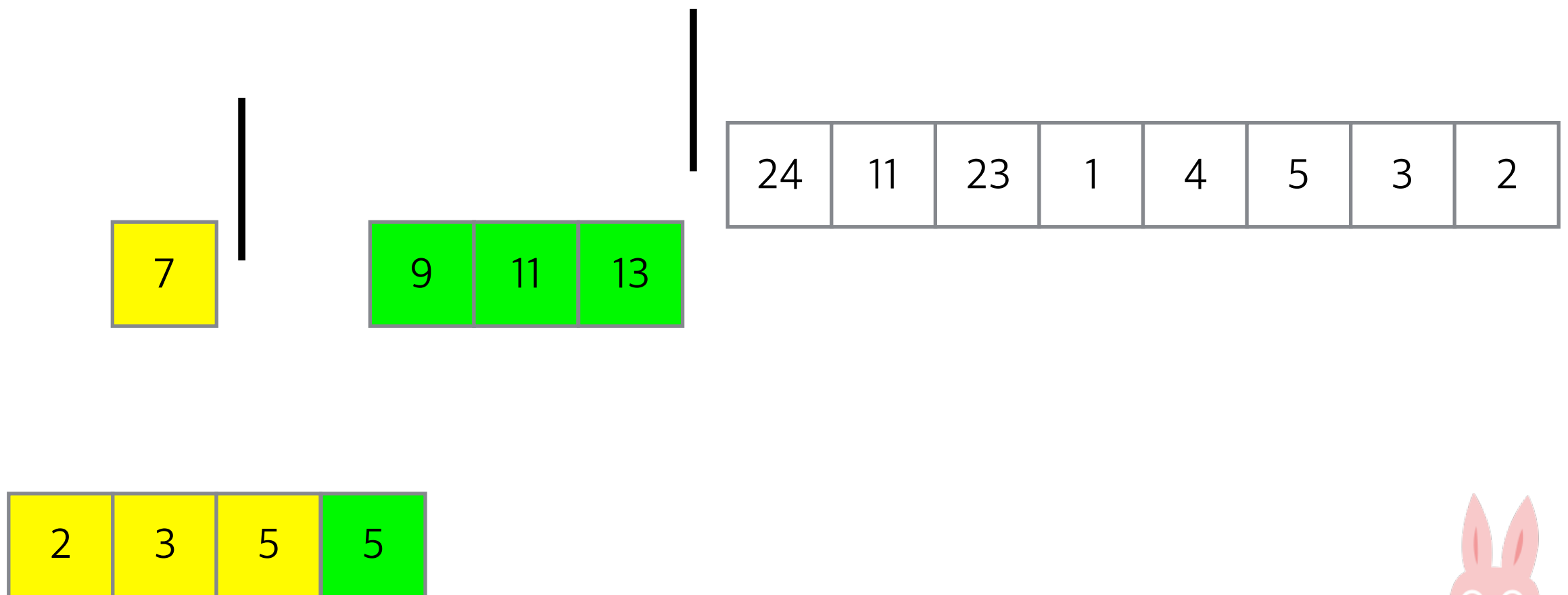
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



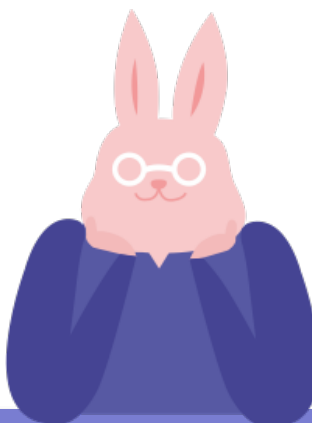
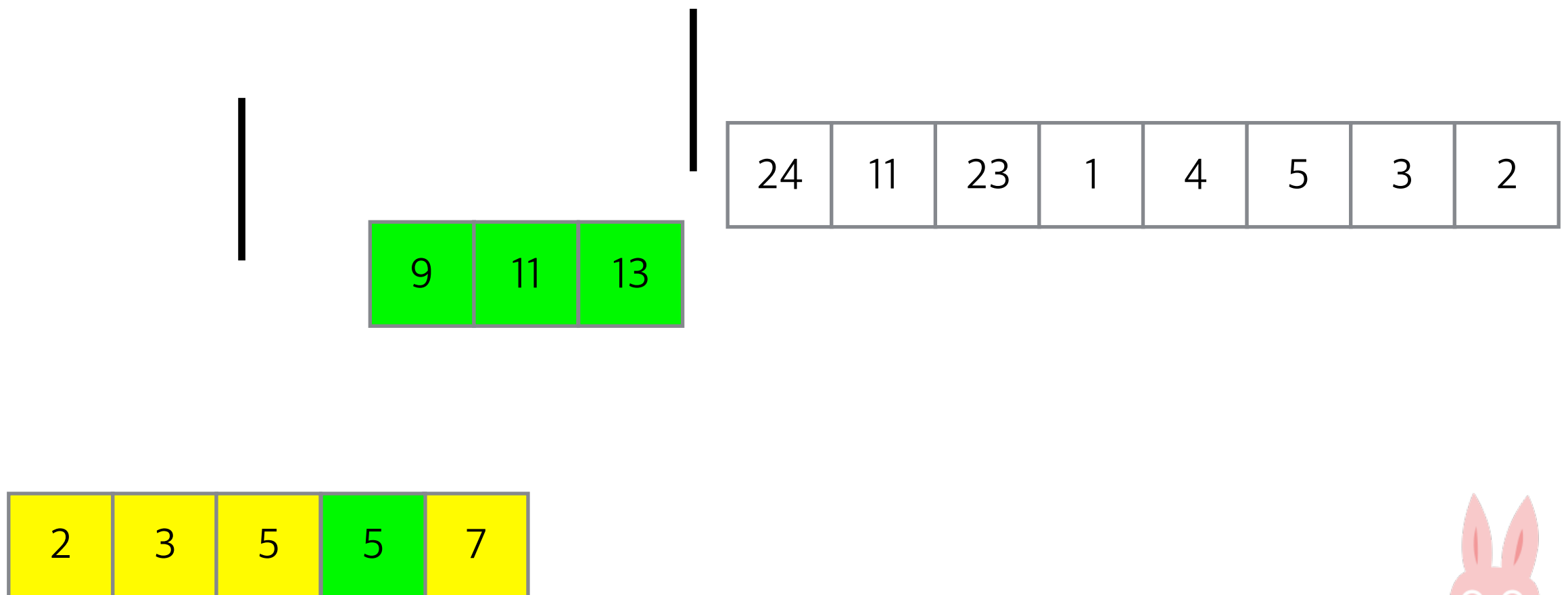
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



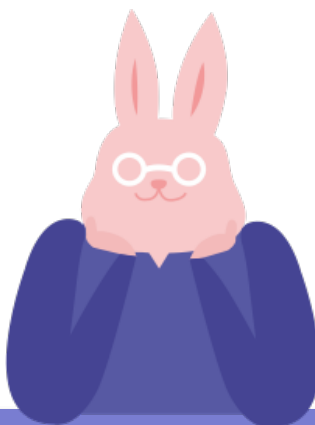
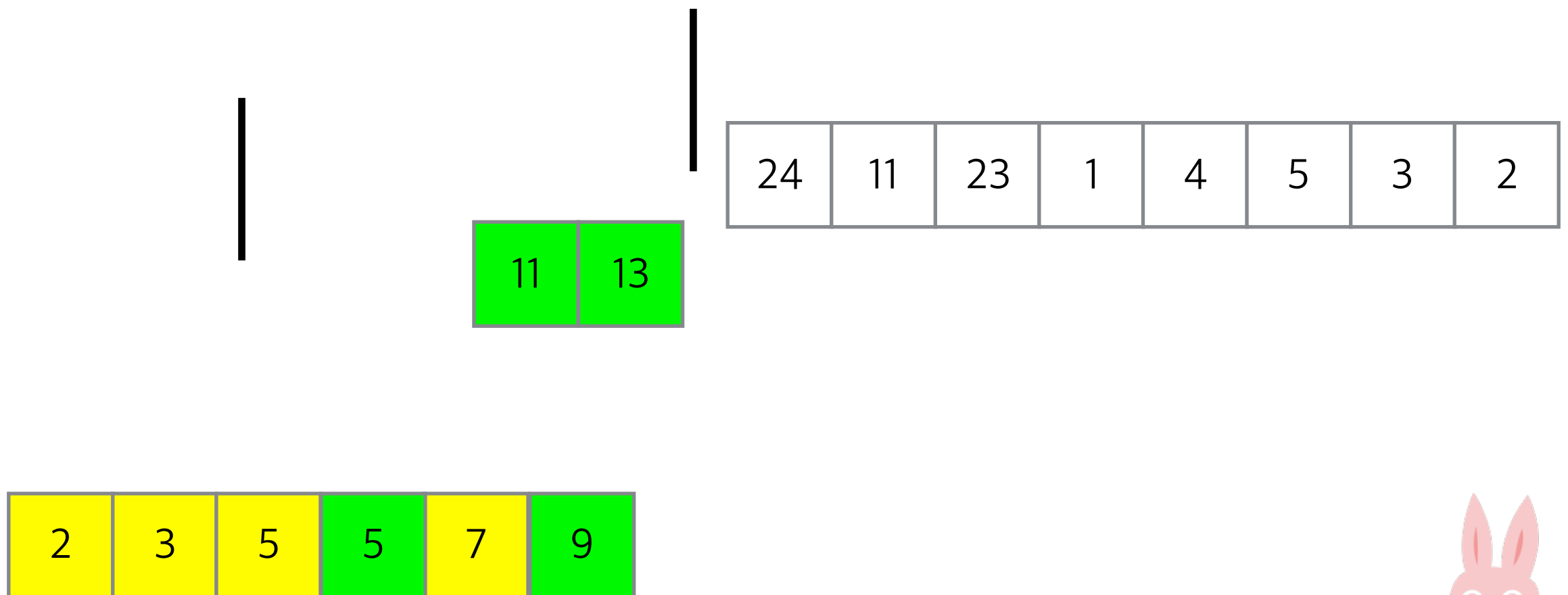
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



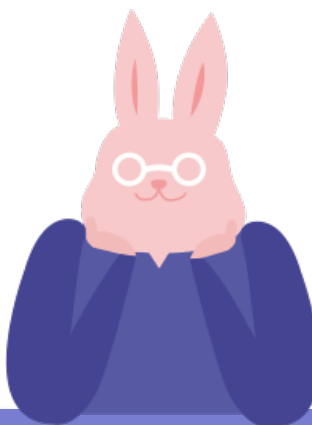
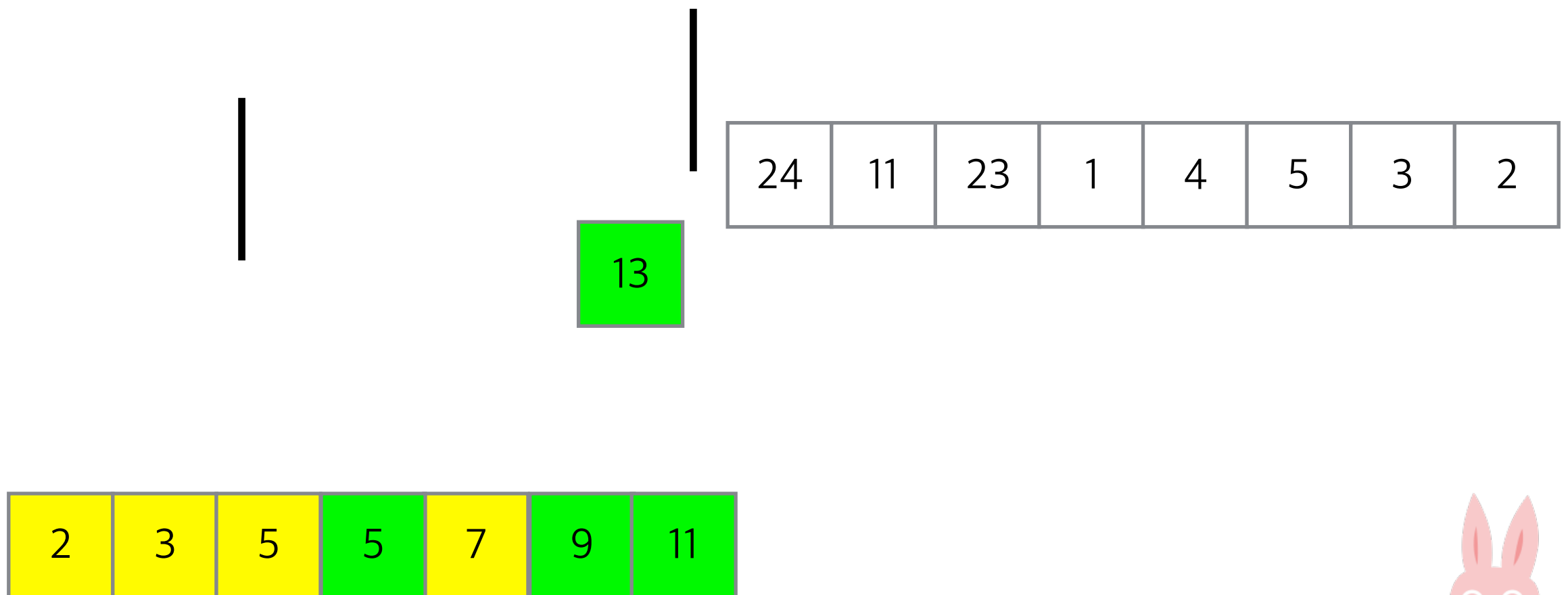
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



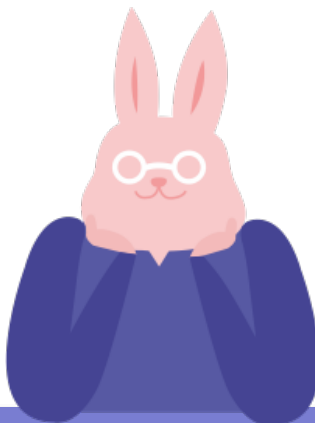
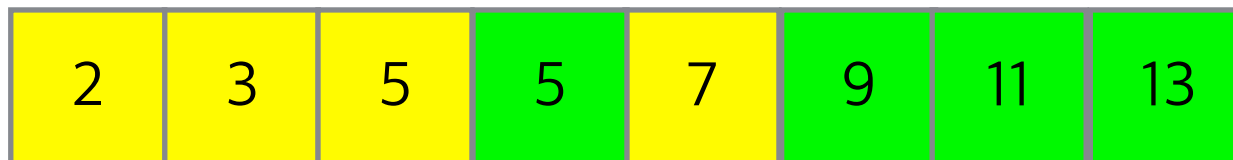
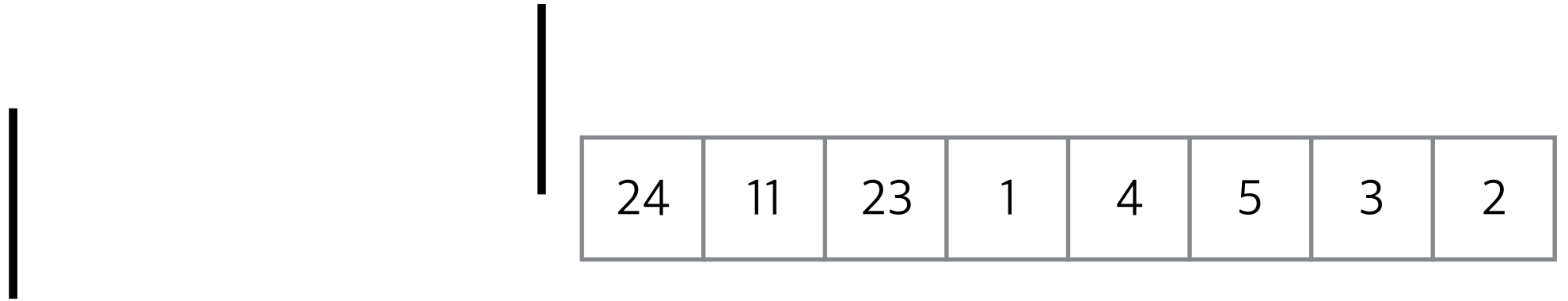
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



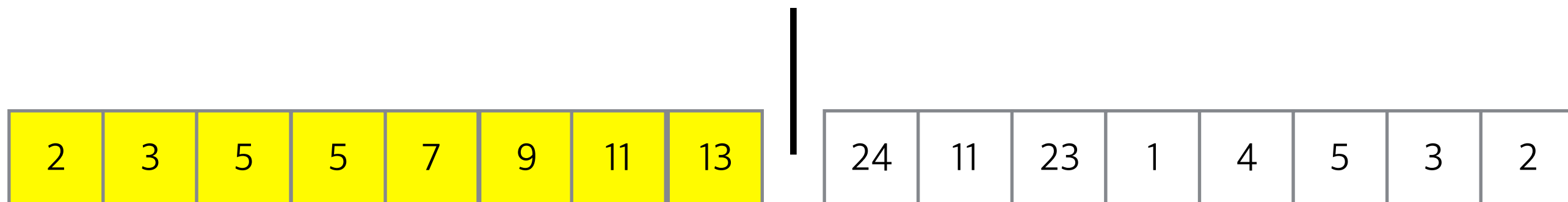
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



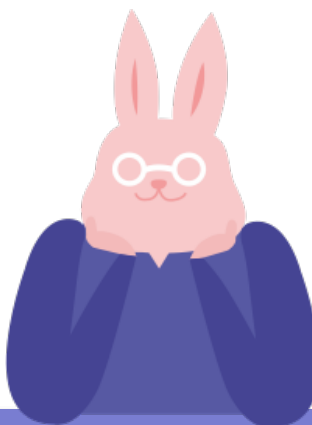
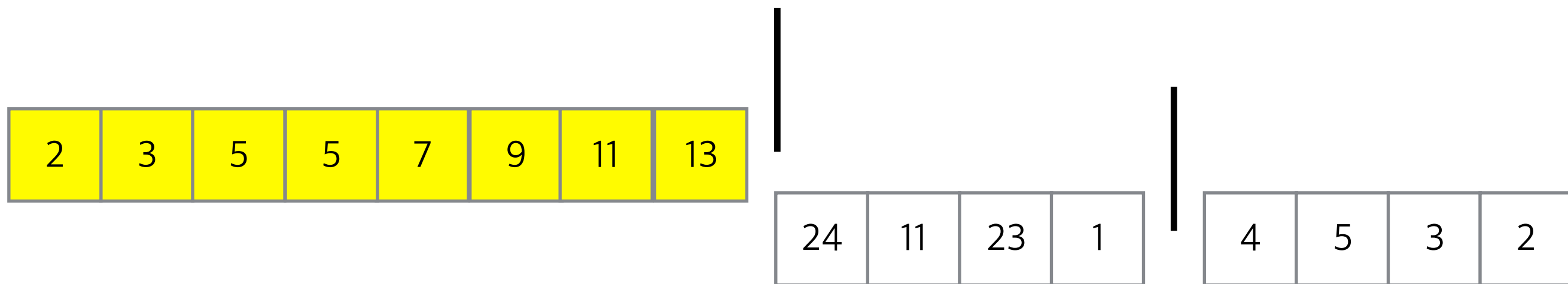
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



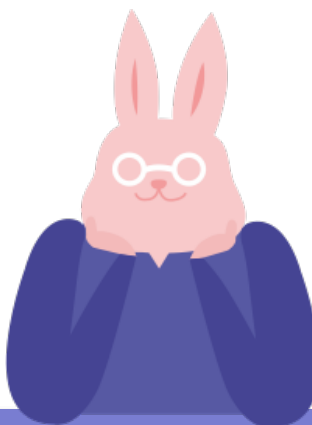
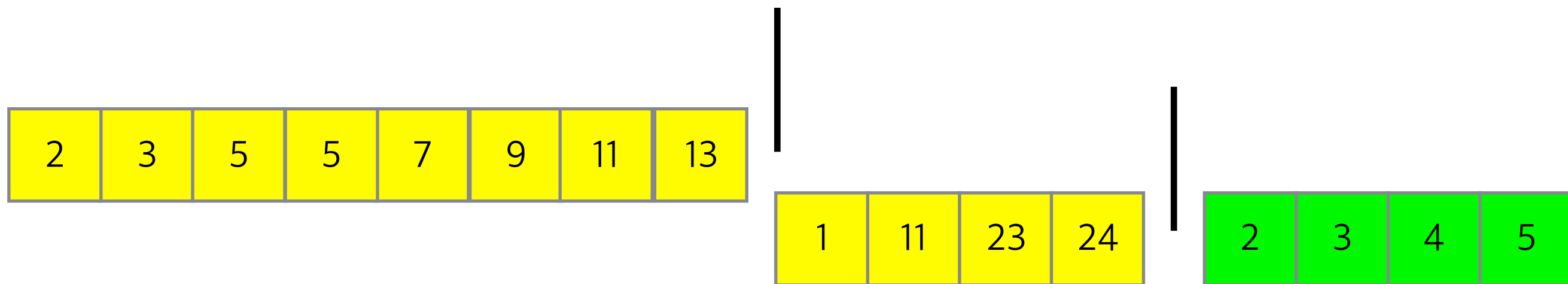
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



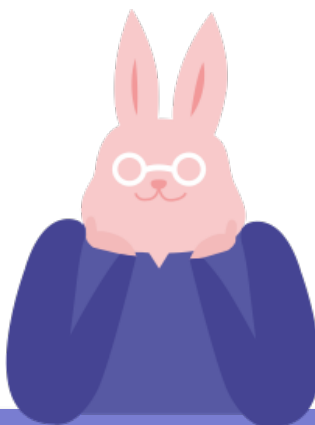
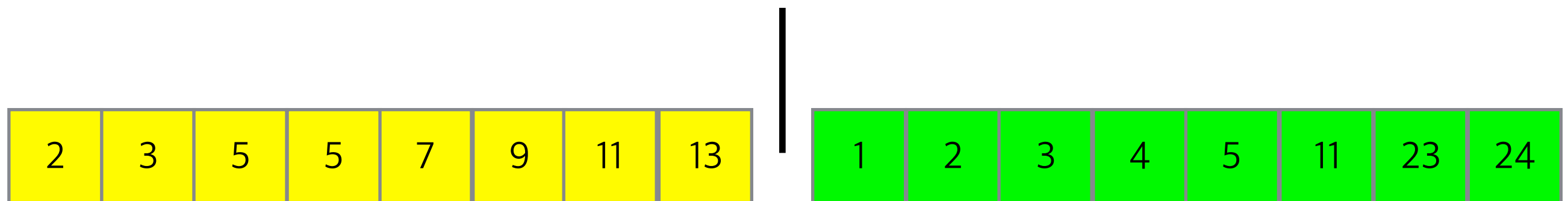
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



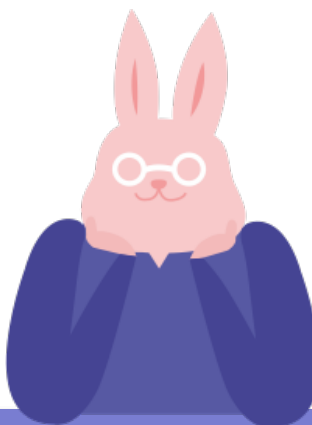
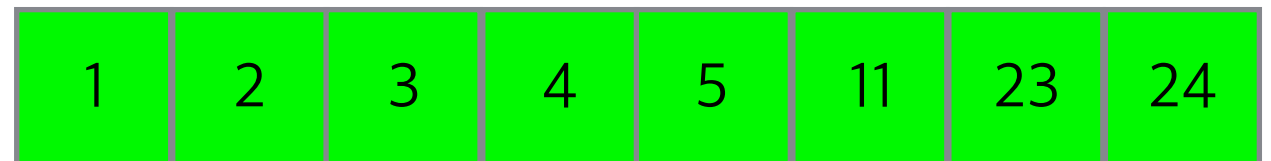
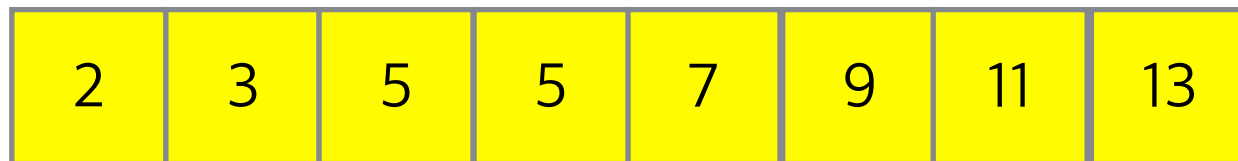
Merge sort

- $O(n \log n)$ 정렬의 가장 대표적인 예



Merge sort

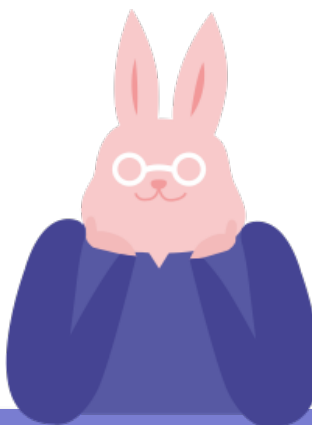
- $O(n \log n)$ 정렬의 가장 대표적인 예



Merge sort

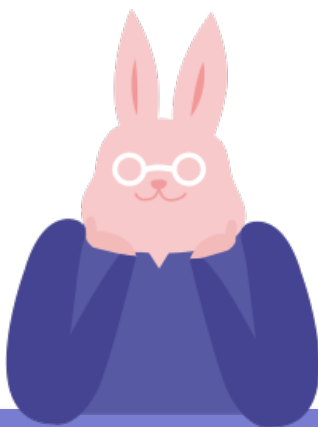
- $O(n \log n)$ 정렬의 가장 대표적인 예

1	2	2	3	3	4	5	5	5	7	9	11	11	13	23	24
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----



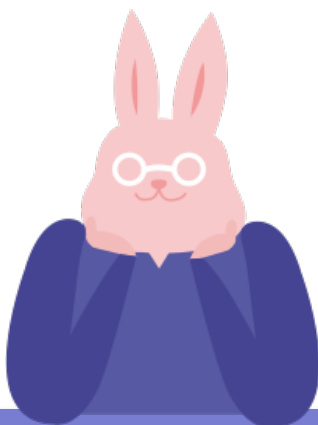
Merge sort : 시간복잡도

- N개의 숫자를 정렬하는 데에 걸리는 시간을 $T(N)$ 이라 하자



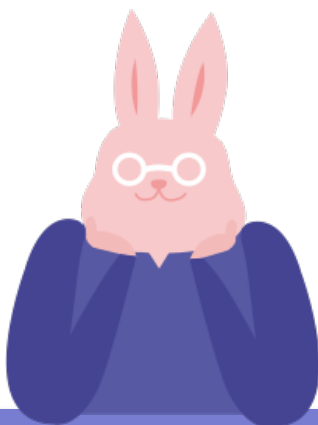
Merge sort : 시간복잡도

- N개의 숫자를 정렬하는 데에 걸리는 시간을 $T(N)$ 이라 하자
- $T(N) = 2 * T(N/2) + O(N)$



Merge sort : 시간복잡도

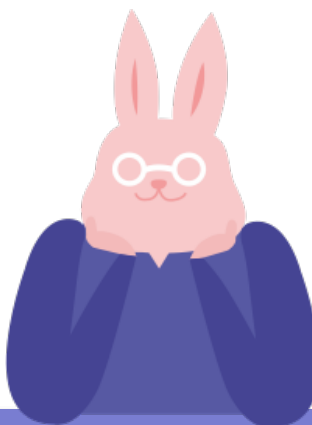
- N개의 숫자를 정렬하는 데에 걸리는 시간을 $T(N)$ 이라 하자
- $T(N) = 2 * T(N/2) + O(N)$



Merge sort : 시간복잡도

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---

$$T(N) = 2 * T(N/2) + O(N)$$



Merge sort : 시간복잡도

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---

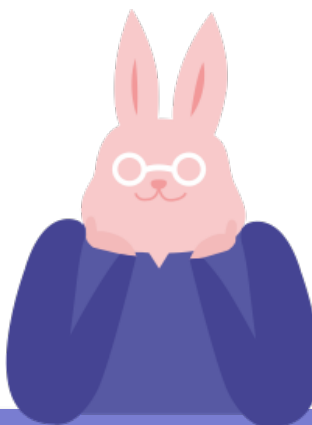
$$T(N) = 2 * T(N/2) + O(N)$$

3	5	7	2	5	9	13	11
---	---	---	---	---	---	----	----

$$T(N/2) = 2 * T(N/4) + \mathbf{O(N/2)}$$

3	5	7	2	5	9	13	11
---	---	---	---	---	---	----	----

$$T(N/2) = 2 * T(N/4) + \mathbf{O(N/2)}$$



Merge sort : 시간복잡도

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---

$$T(N) = 2 * T(N/2) + O(N)$$

3	5	7	2	5	9	13	11
---	---	---	---	---	---	----	----

3	5	7	2	5	9	13	11
---	---	---	---	---	---	----	----

$$T(N/2) = 2 * T(N/4) + \mathbf{O(N/2)}$$

$$T(N/2) = 2 * T(N/4) + \mathbf{O(N/2)}$$

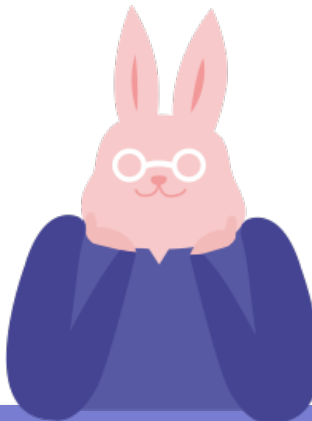
3	5	7	2
---	---	---	---

5	9	13	11
---	---	----	----

3	5	7	2
---	---	---	---

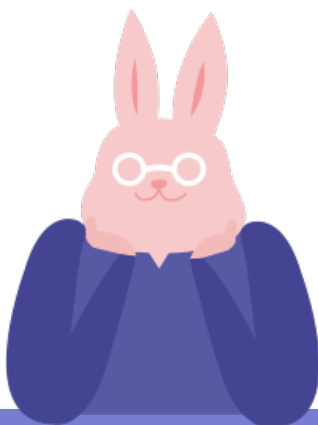
5	9	13	11
---	---	----	----

$$T(N/4) = \dots + \mathbf{O(N/4)} \quad T(N/4) = \dots + \mathbf{O(N/4)} \quad T(N/4) = \dots + \mathbf{O(N/4)} \quad T(N/4) = \dots + \mathbf{O(N/4)}$$



Merge sort : 시간복잡도

- N개의 숫자를 정렬하는 데에 걸리는 시간을 $T(N)$ 이라 하자
- $T(N) = 2 * T(N/2) + O(N)$
- 각 층이 $O(N)$, 총 층의 개수가 $\log N$ 개 : $O(N \log N)$



[활동문제 1] Merge Sort

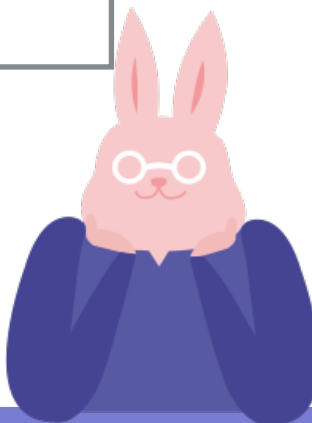
- n개의 숫자를 오름차순으로 정렬

입력의 예

2 5 3 4 1

출력의 예

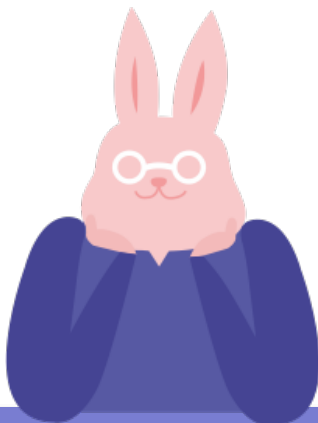
1 2 3 4 5



Quick Sort

- 또 다른 $O(n \log n)$ 정렬의 대표적인 예

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---

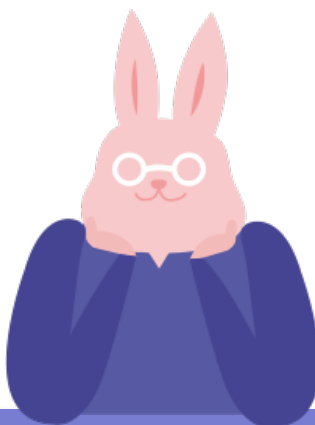


Quick Sort

- 또 다른 $O(n \log n)$ 정렬의 대표적인 예

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---

Pivot!



Quick Sort

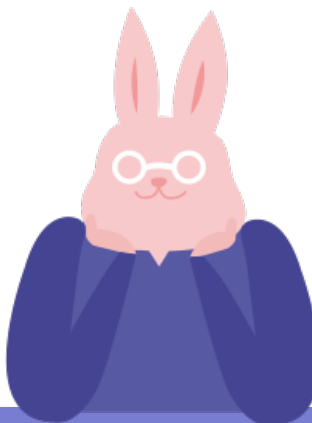
- 또 다른 $O(n \log n)$ 정렬의 대표적인 예

3	5	7
---	---	---

5	9	13	11	24	11	23
---	---	----	----	----	----	----

4	5
---	---

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



Quick Sort

- 또 다른 $O(n \log n)$ 정렬의 대표적인 예

3

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

5	7	5	9	13	11	24	11	23	4	5
---	---	---	---	----	----	----	----	----	---	---



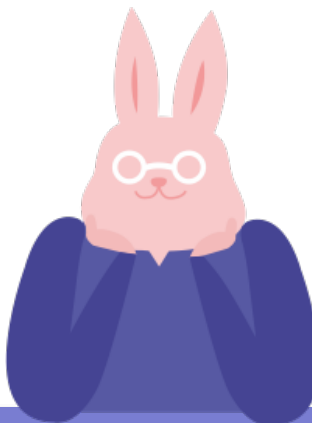
Quick Sort

- 또 다른 $O(n \log n)$ 정렬의 대표적인 예

3

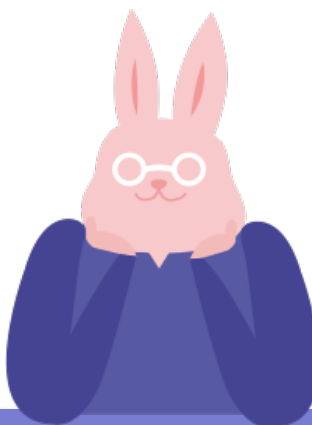
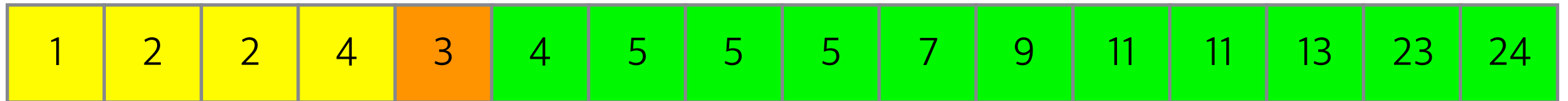
1	2	2	3
---	---	---	---

4	5	5	5	7	9	11	11	13	23	24
---	---	---	---	---	---	----	----	----	----	----



Quick Sort

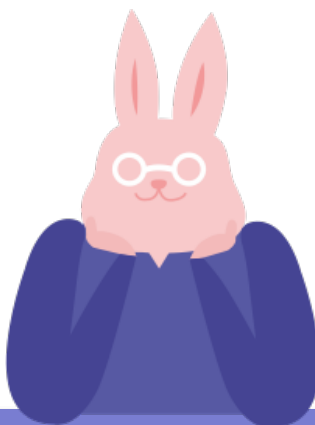
- 또 다른 $O(n \log n)$ 정렬의 대표적인 예



Quick Sort

- 또 다른 $O(n \log n)$ 정렬의 대표적인 예

3	5	7	2	5	9	13	11	24	11	23	1	4	5	3	2
---	---	---	---	---	---	----	----	----	----	----	---	---	---	---	---



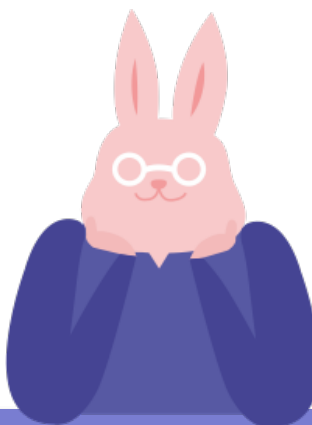
Quick Sort

- 또 다른 $O(n \log n)$ 정렬의 대표적인 예

3

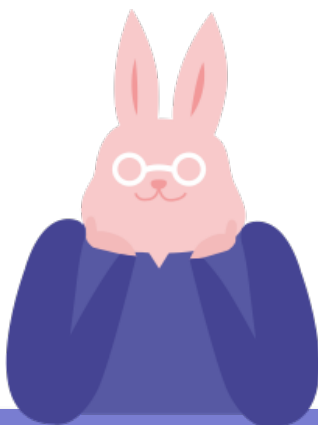
1	3	2	2	5	7	5	9	13	11	24	11	23	4	5
---	---	---	---	---	---	---	---	----	----	----	----	----	---	---

각각을 Quick Sort로 정렬!



Quick Sort : 시간복잡도

- N개의 숫자를 정렬하는 데에 걸리는 시간을 $T(N)$ 이라 하자
- $T(N) = 2 * T(N/2) + O(N)$
(대충 절반으로 나누어진다고 가정하면)
- $O(N \log N)$



[활동문제 2] Quick Sort

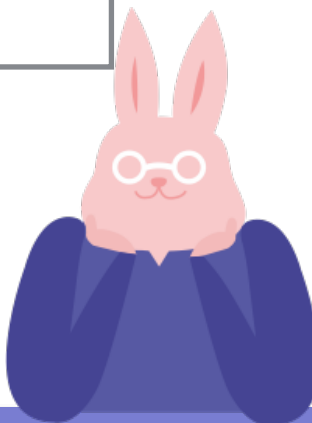
- n개의 숫자를 오름차순으로 정렬

입력의 예

2 5 3 4 1

출력의 예

1 2 3 4 5



[실습문제 3-3] 최대구간의 합 구하기 3

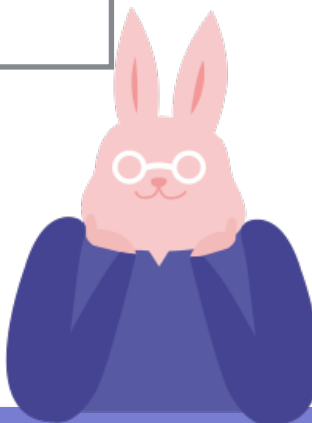
- n개의 숫자 중에서 연속 부분 최대합을 출력

입력의 예

```
1 2 3 4 -100 1
```

출력의 예

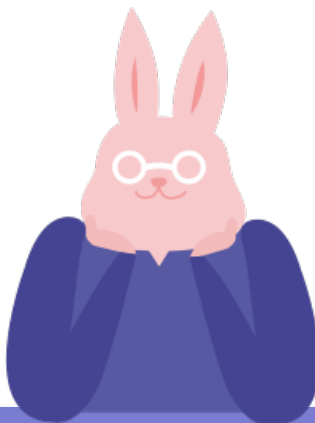
```
10
```



가장 단순한 방법

- $O(N^3)$

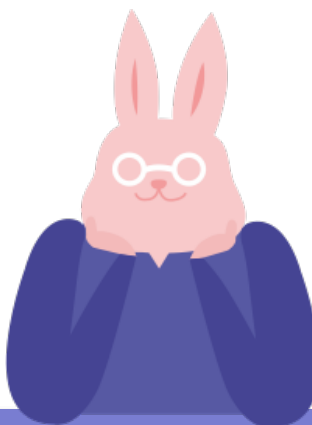
2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----



구간 합 $O(1)$ 에 구하기

- $S[n]$: $arr[0] \sim arr[n]$ 까지의 합
- $S[n]$ 을 미리 한번만 구해놓으면 구간 합은 $O(1)$ 에 가능

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

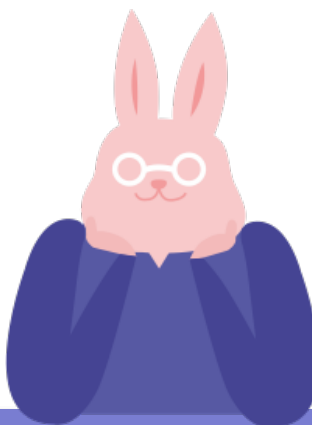


구간 합 $O(1)$ 에 구하기

- $S[n]$: $arr[0] \sim arr[n]$ 까지의 합
- $S[n]$ 을 미리 한번만 구해놓으면 구간 합은 $O(1)$ 에 가능

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

S	2	3	1	6	-4	-1	1	6	3	10	19	9
---	---	---	---	---	----	----	---	---	---	----	----	---



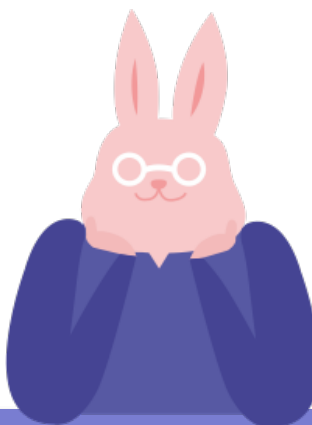
구간 합 $O(1)$ 에 구하기

- $S[n]$: $arr[0] \sim arr[n]$ 까지의 합
- $S[n]$ 을 미리 한번만 구해놓으면 구간 합은 $O(1)$ 에 가능

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

S	2	3	1	6	-4	-1	1	6	3	10	19	9
---	---	---	---	---	----	----	---	---	---	----	----	---

$arr[3] \sim arr[7]$ 까지의 합은 ?



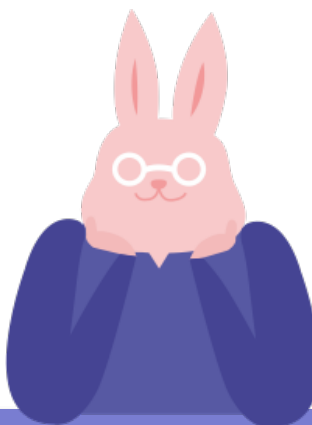
구간 합 $O(1)$ 에 구하기

- $S[n]$: $arr[0] \sim arr[n]$ 까지의 합
- $S[n]$ 을 미리 한번만 구해놓으면 구간 합은 $O(1)$ 에 가능

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

S	2	3	1	6	-4	-1	1	6	3	10	19	9
---	---	---	---	---	----	----	---	---	---	----	----	---

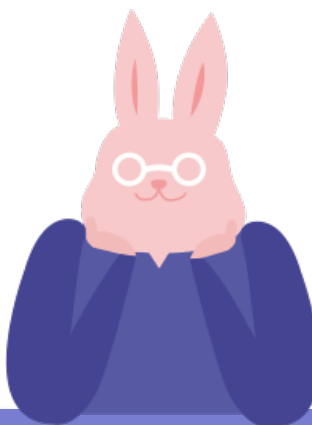
$arr[3] \sim arr[7]$ 까지의 합은 ? $S[7] - S[2]$



가장 (까진 아니지만) 단순한 방법

- $O(N^2)$

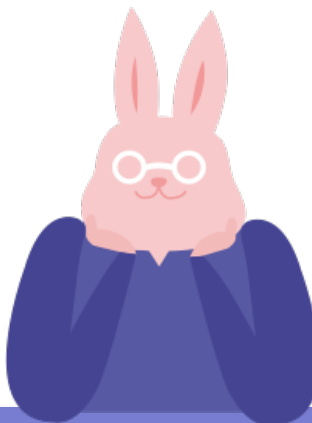
2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----



분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자

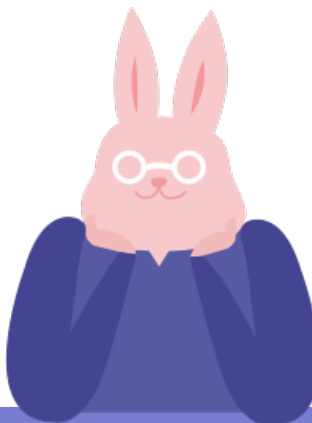
2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----



분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자

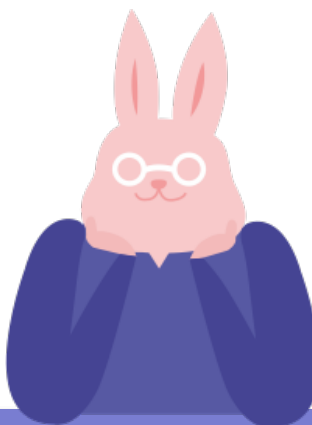
2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----



분할정복법을 활용

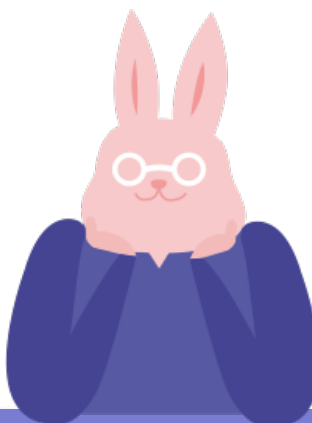
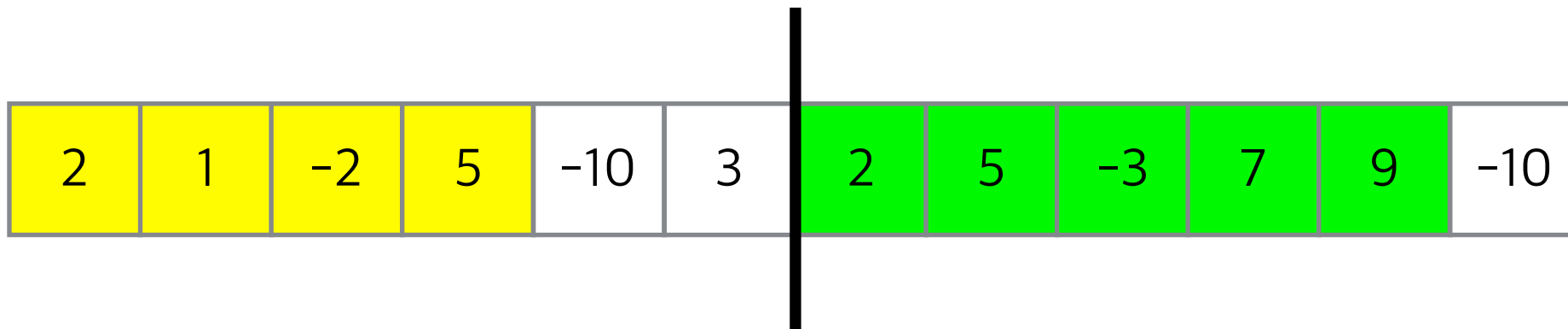
- 우선 절반으로 나누어 각각을 구해보자

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----



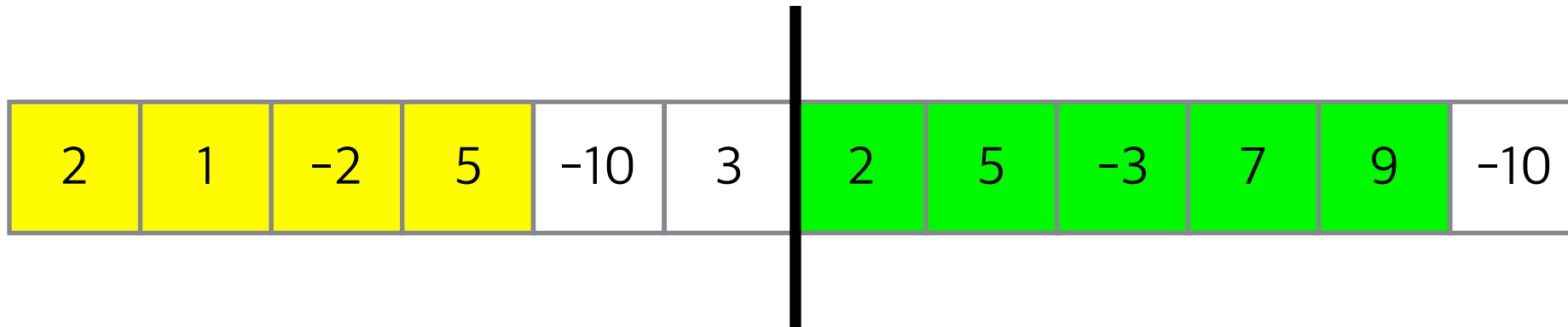
분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자

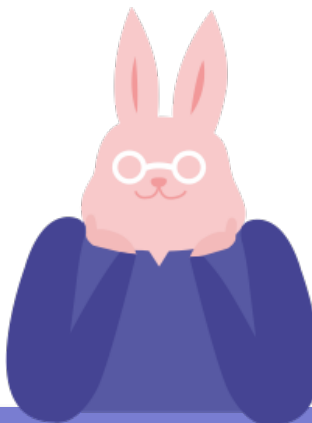


분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자

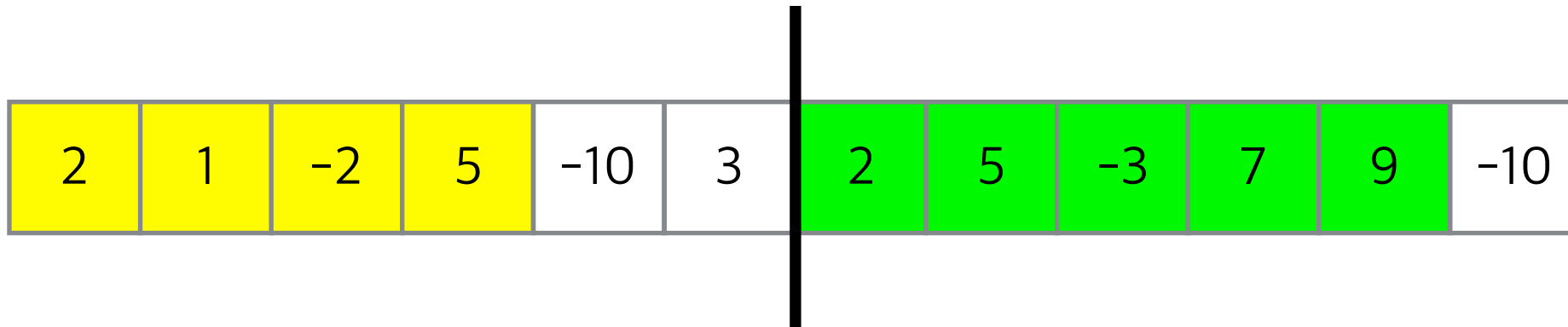


- 고려하지 않은 경우는 ?

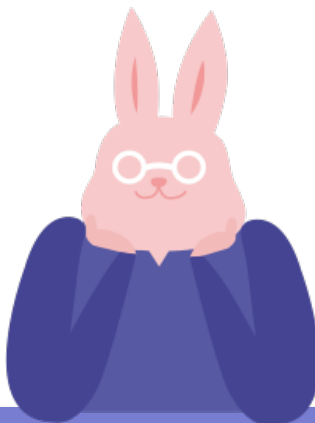


분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자

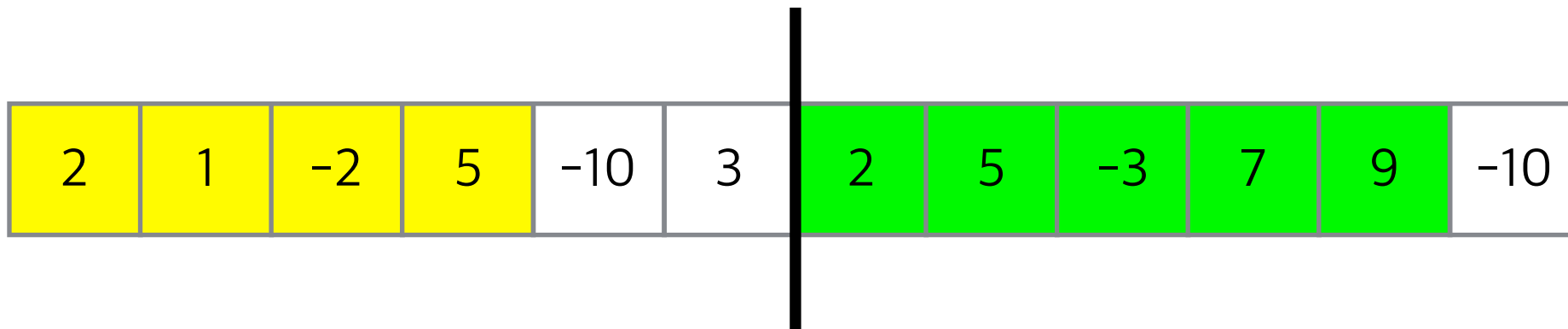


- 고려하지 않은 경우는 ?
 - 자른 지점을 포함하는 연속 부분!

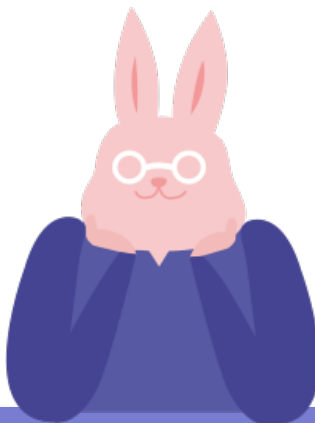


분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자

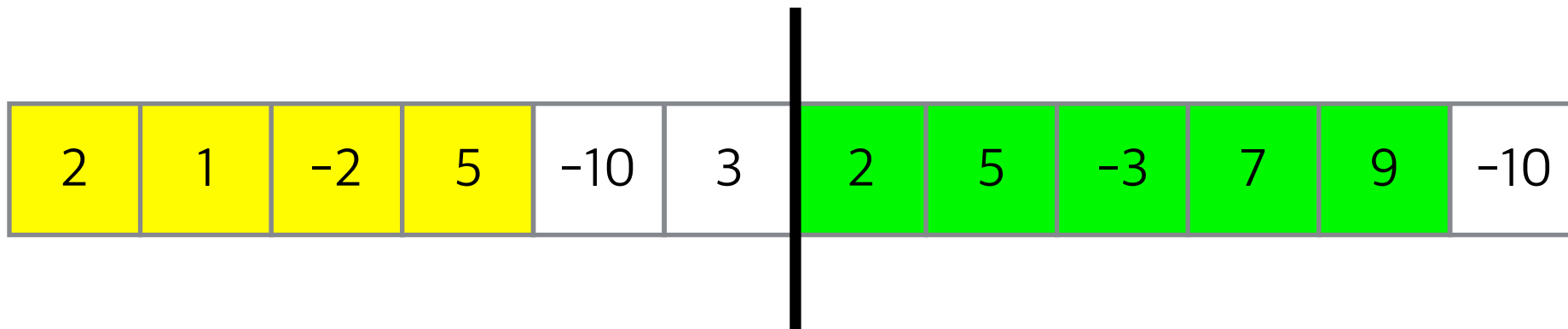


- 자른 지점을 포함하는 연속 부분의 최대 합은 어떻게 구할까?

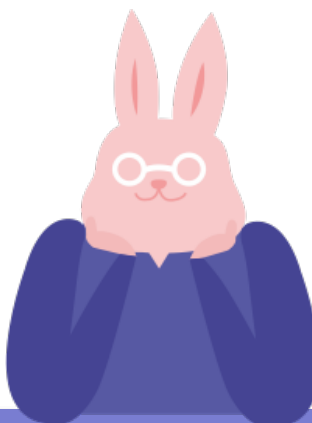


분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자

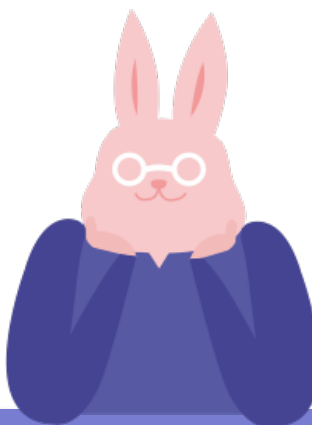
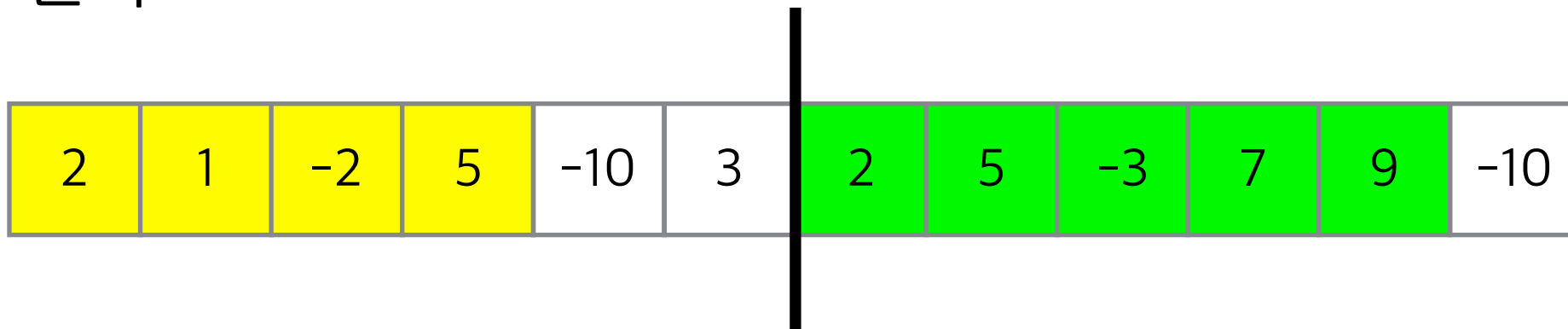


- 자른 지점을 포함하는 연속 부분의 최대 합은 어떻게 구할까?
 - Idea : 왼쪽과 오른쪽을 독립적으로 생각하자



분할정복법을 활용

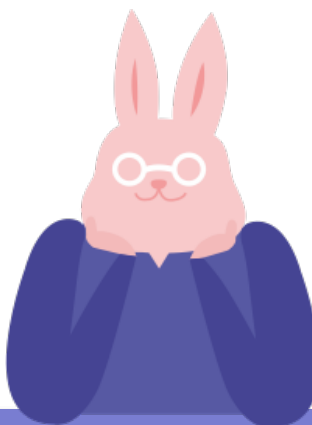
- 우선 절반으로 나누어 각각을 구해보자
 - 왼쪽



분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자
 - 왼쪽

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

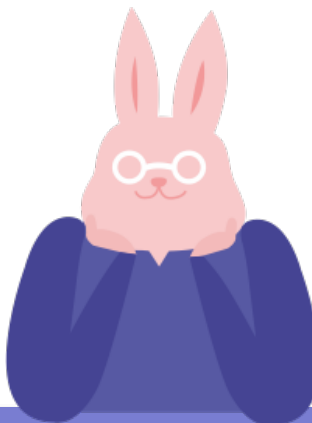


분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자
 - 왼쪽

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

					3
				-10	3
			5	-10	3
		-2	5	-10	3
	1	-2	5	-10	3
2	1	-2	5	-10	3

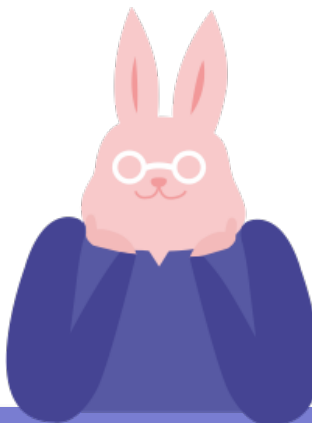


분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자
 - 왼쪽

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

					3	3
				-10	3	-7
			5	-10	3	-2
		-2	5	-10	3	-4
	1	-2	5	-10	3	-3
2	1	-2	5	-10	3	-1



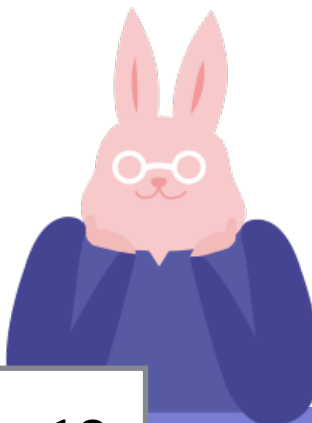
분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자
 - 오른쪽

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

					3	3				2			
				-10	3	-7			2	5			
		5	-10	3	-2			2	5	-3			
	-2	5	-10	3	-4			2	5	-3	7		
	1	-2	5	-10	3	-3		2	5	-3	7	9	
2	1	-2	5	-10	3	-1		2	5	-3	7	9	-10

113

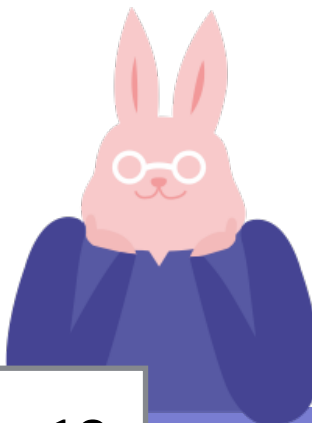


분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자
 - 오른쪽

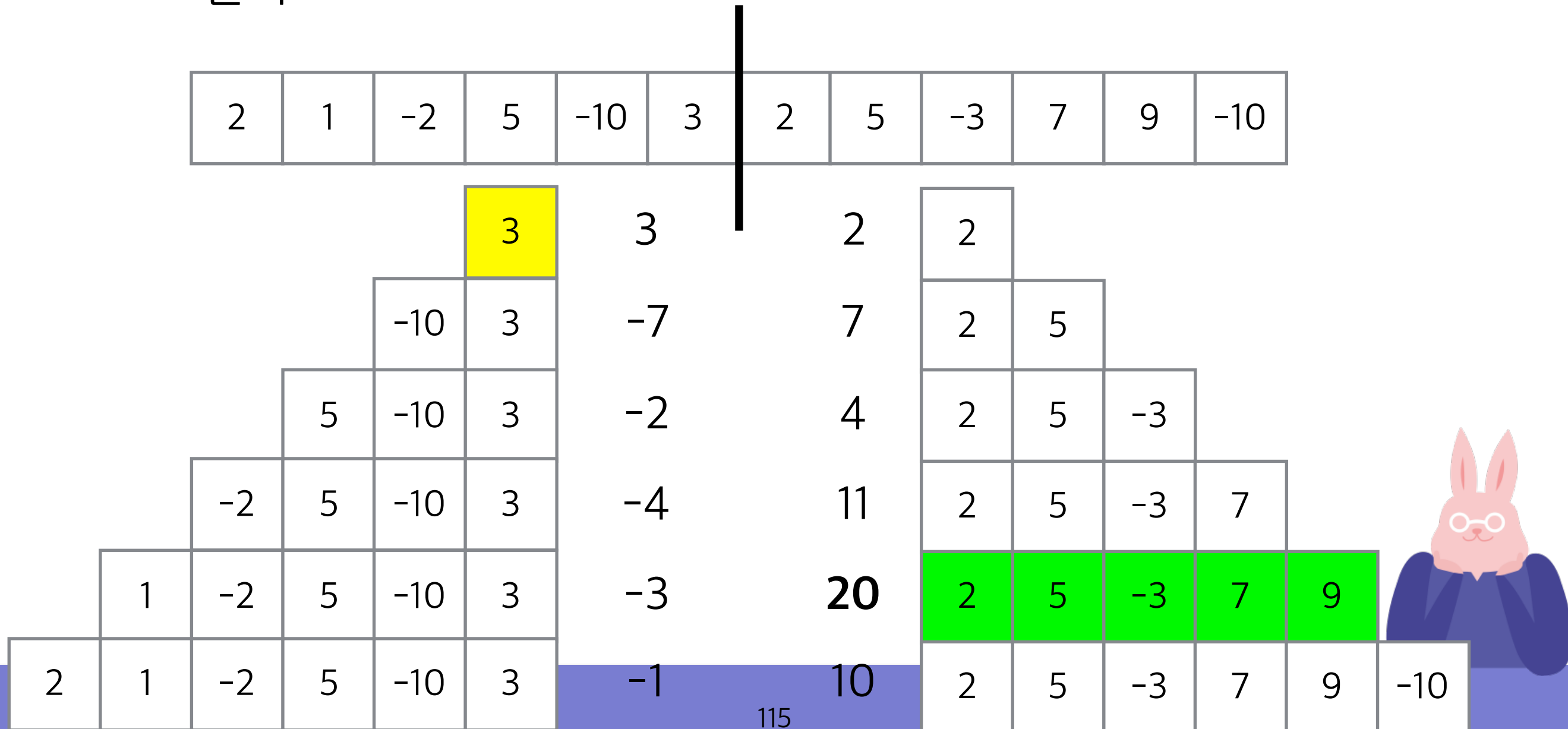
2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----

					3		3		2		2						
					-10	3		-7		7		2	5				
				5	-10	3		-2		4		2	5	-3			
		-2	5	-10	3			-4		11		2	5	-3	7		
	1	-2	5	-10	3			-3		20		2	5	-3	7	9	
2	1	-2	5	-10	3			-1		10		2	5	-3	7	9	-10



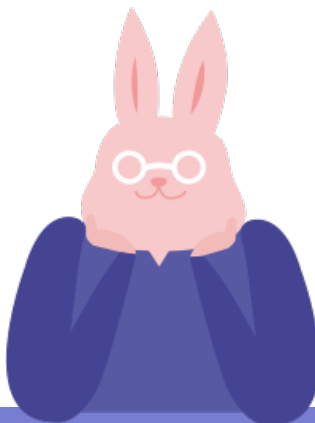
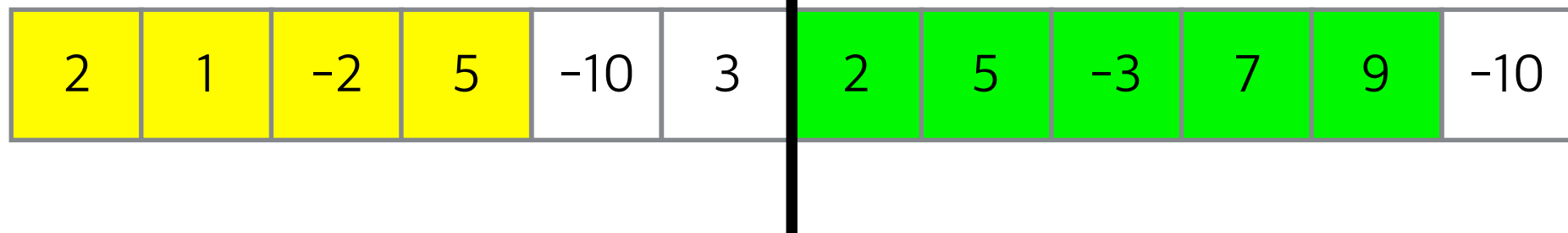
분할정복법을 활용

- 우선 절반으로 나누어 각각을 구해보자
 - 결과!



분할정복법을 활용

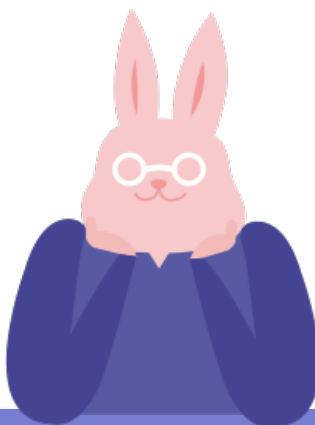
- 모든 경우를 고려했음
 - 왼쪽만 포함하는 경우,
 - 오른쪽만 포함하는 경우,
 - 자른 자리를 포함하는 경우



분할정복법을 활용

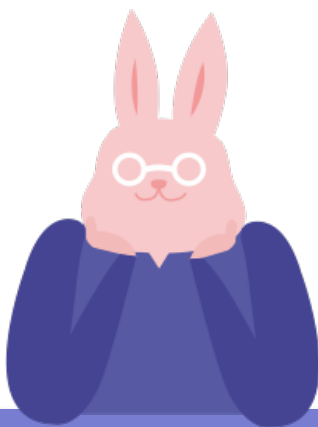
- 최댓값!

2	1	-2	5	-10	3	2	5	-3	7	9	-10
---	---	----	---	-----	---	---	---	----	---	---	-----



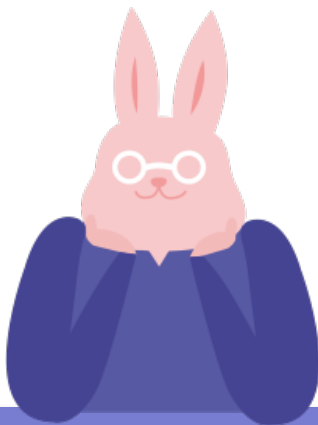
시간복잡도

- $T(N) = 2 * T(N/2) + O(N)$
- $O(N \log N)$



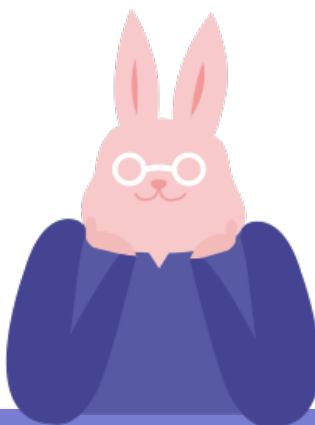
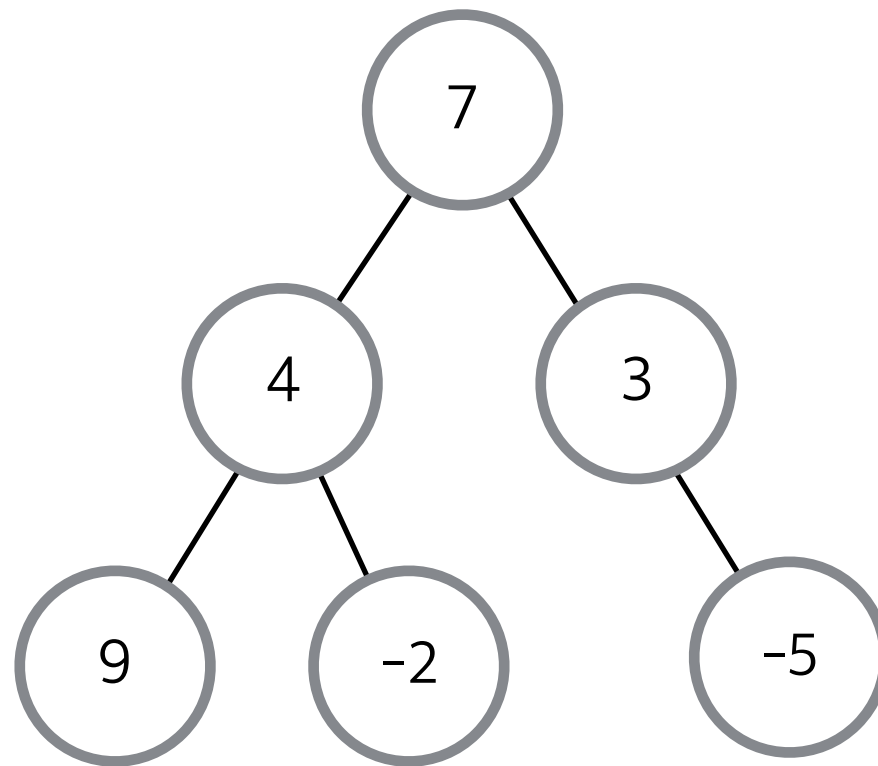
분할정복법 요약

- 고려해야 하는 경우를 줄일 수 있다
- 보통 분할정복법 문제는 어려우니, 예제 풀이는 꼭 외우자



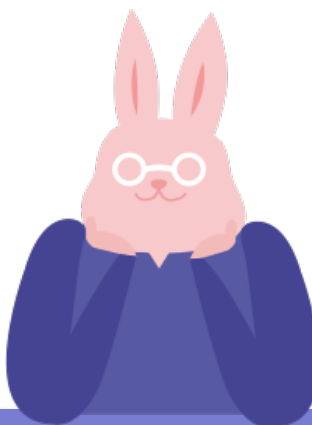
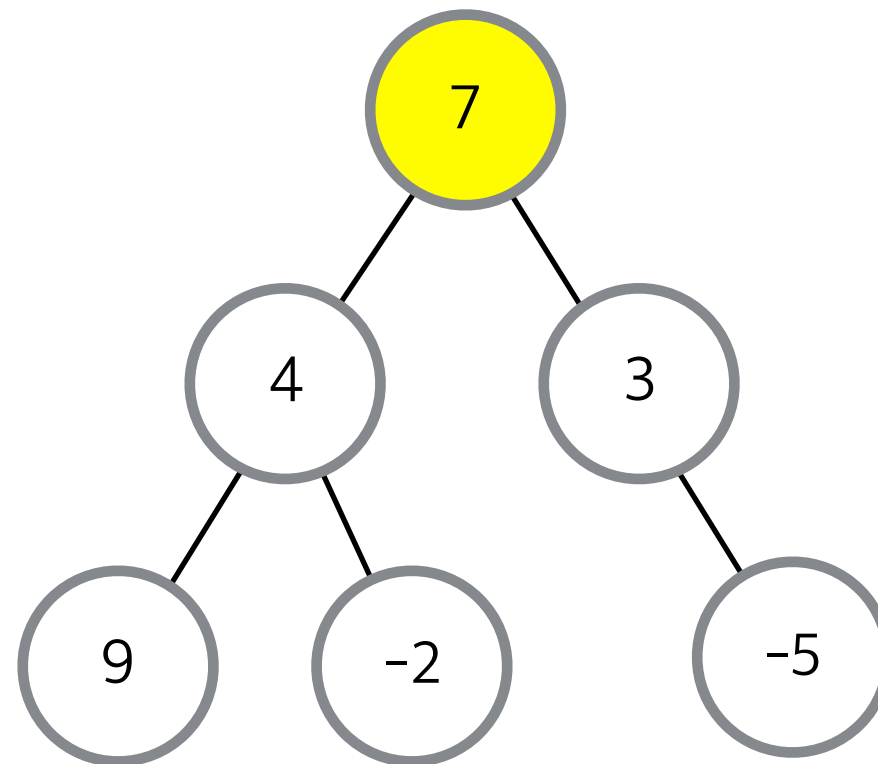
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



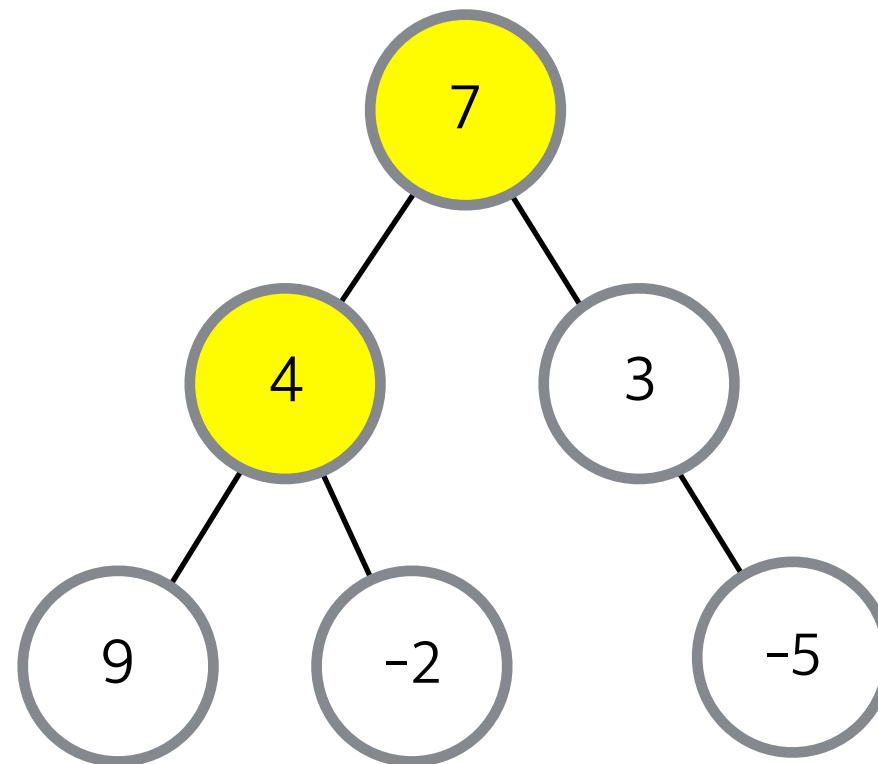
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



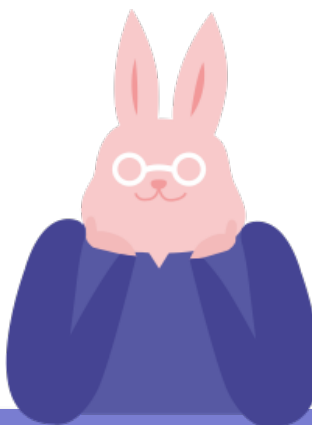
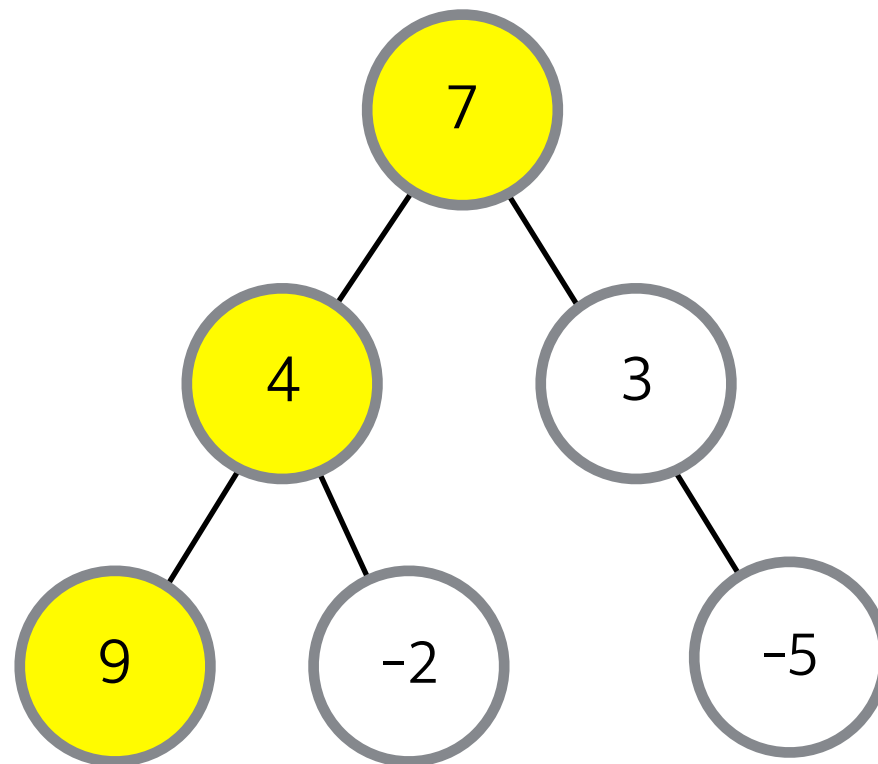
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



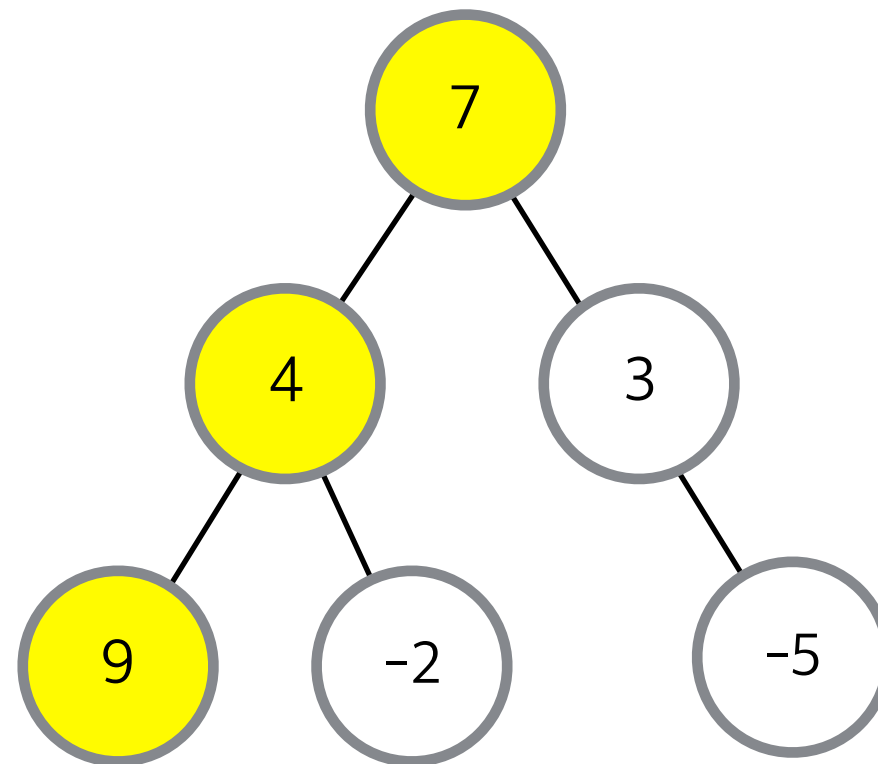
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



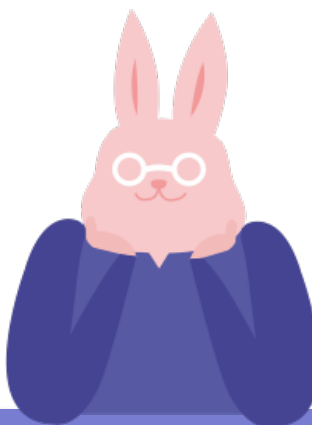
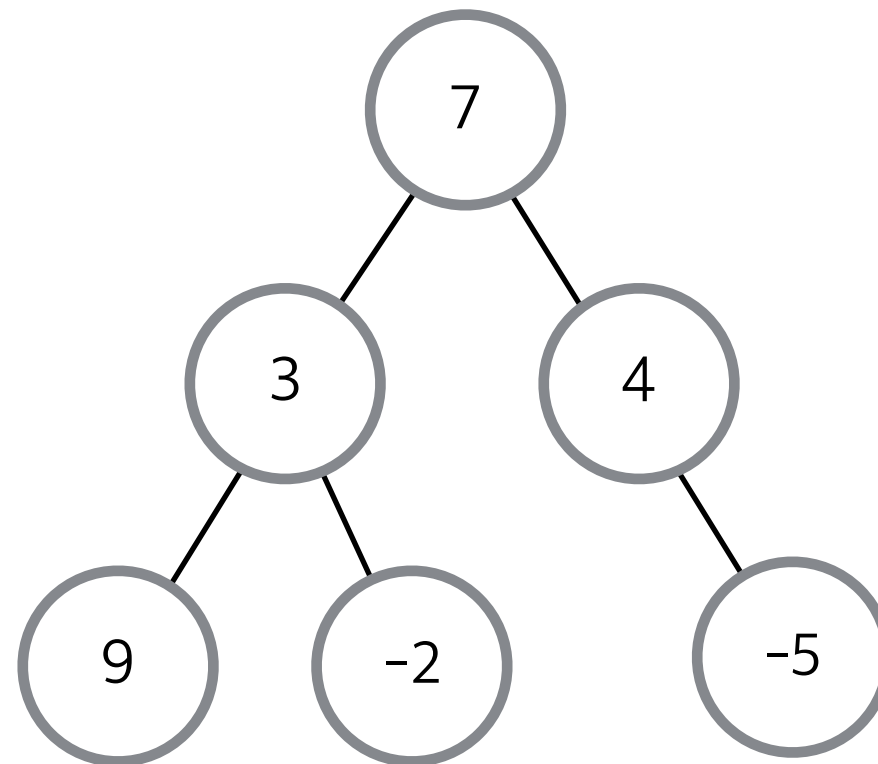
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



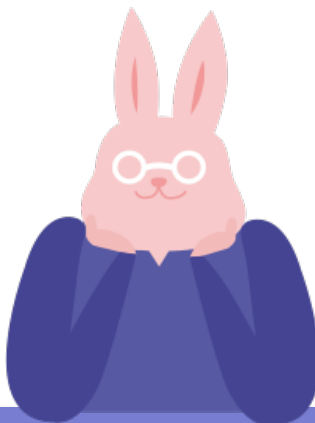
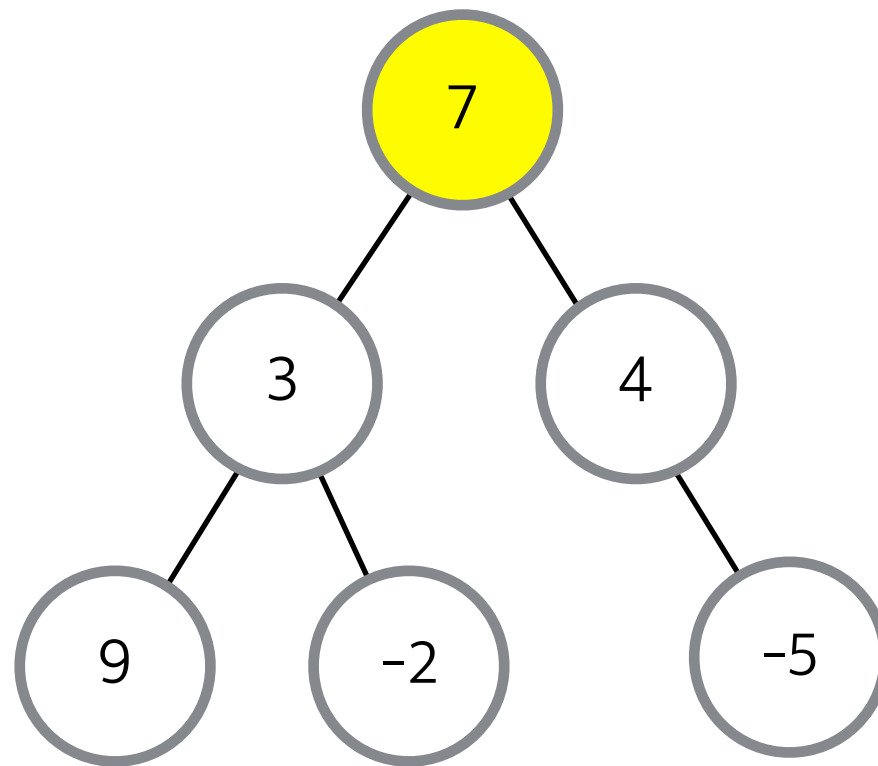
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



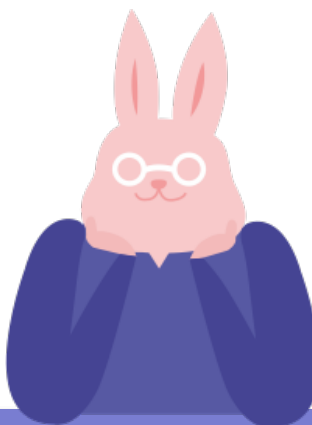
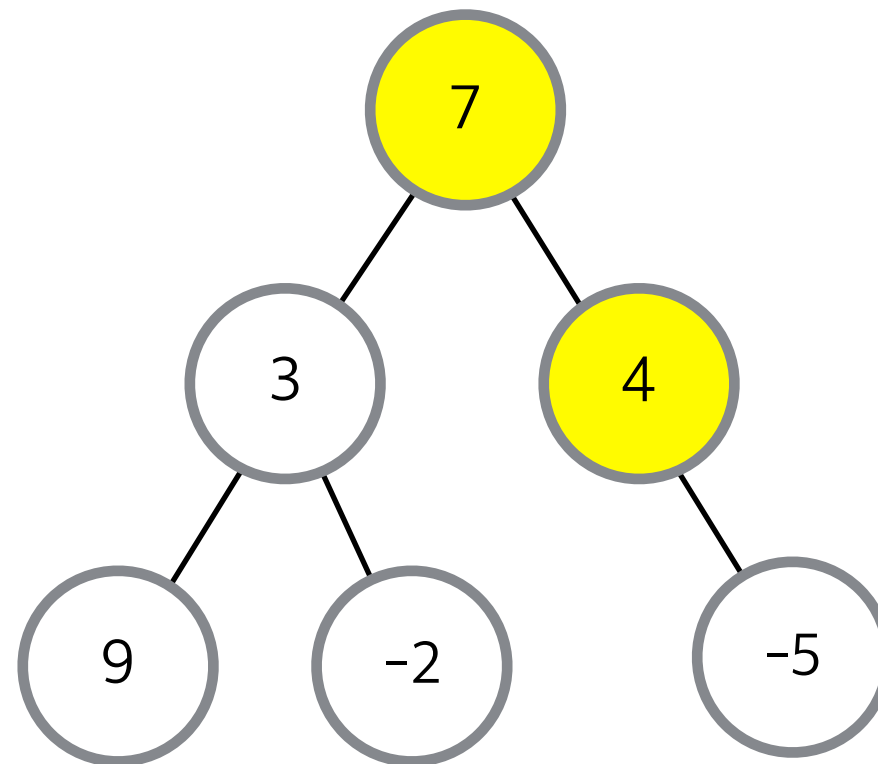
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



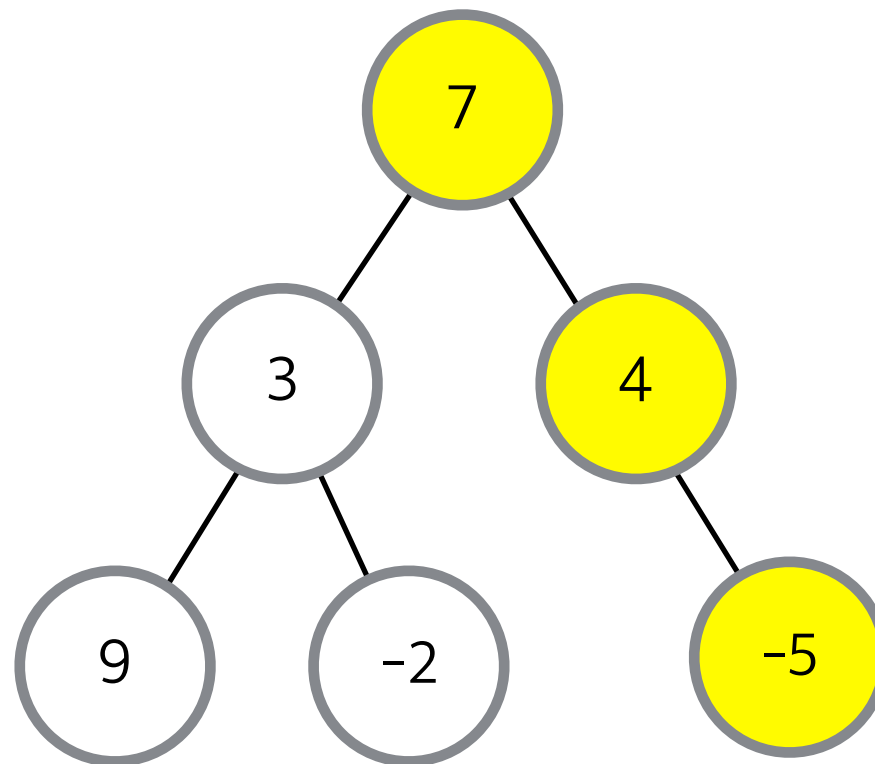
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



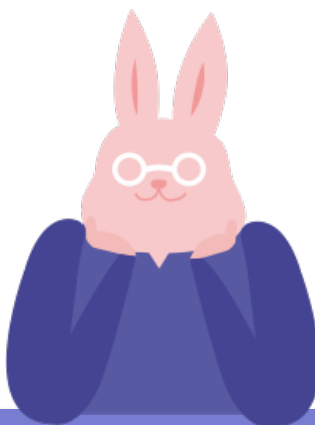
탐욕적 기법 (Greedy)

- 현재 선택할 수 있는 경우 중에서 항상 최적의 선택을 하자
- 예제 : Root에서 내려오면서, 합을 최대화 하라



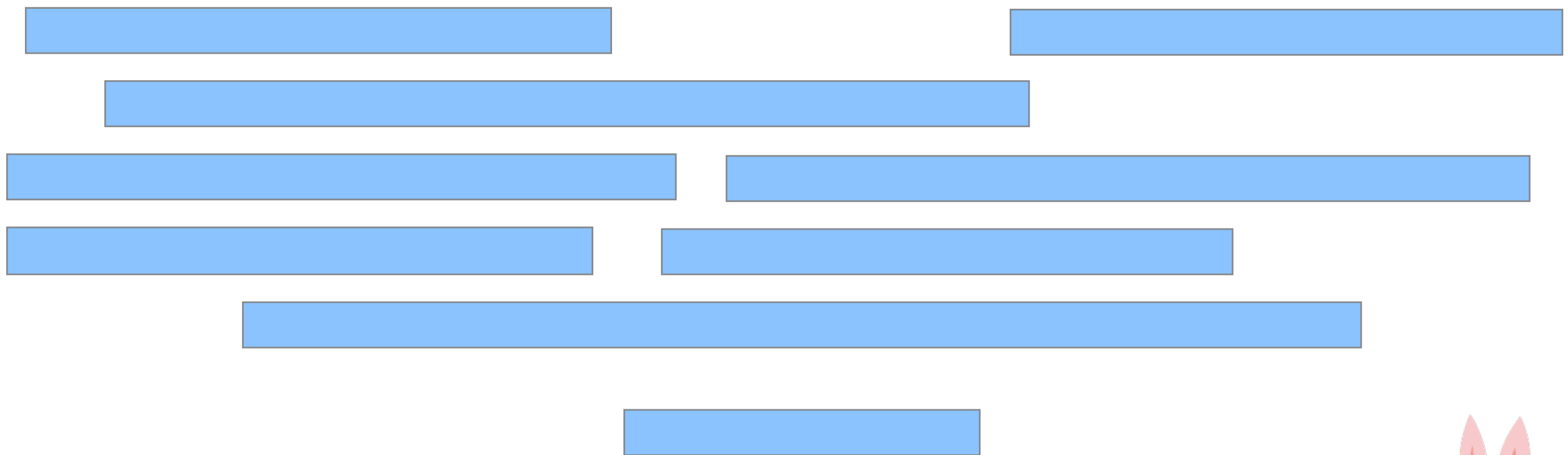
탐욕적 기법 (Greedy)

- 탐욕적 기법으로 문제가 풀린다는 것을 **증명**해야 함



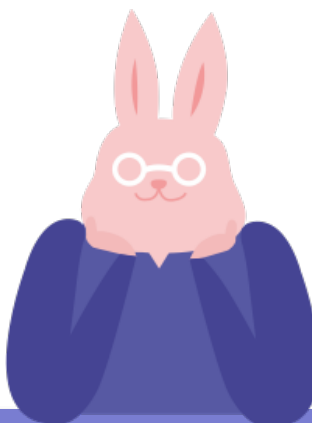
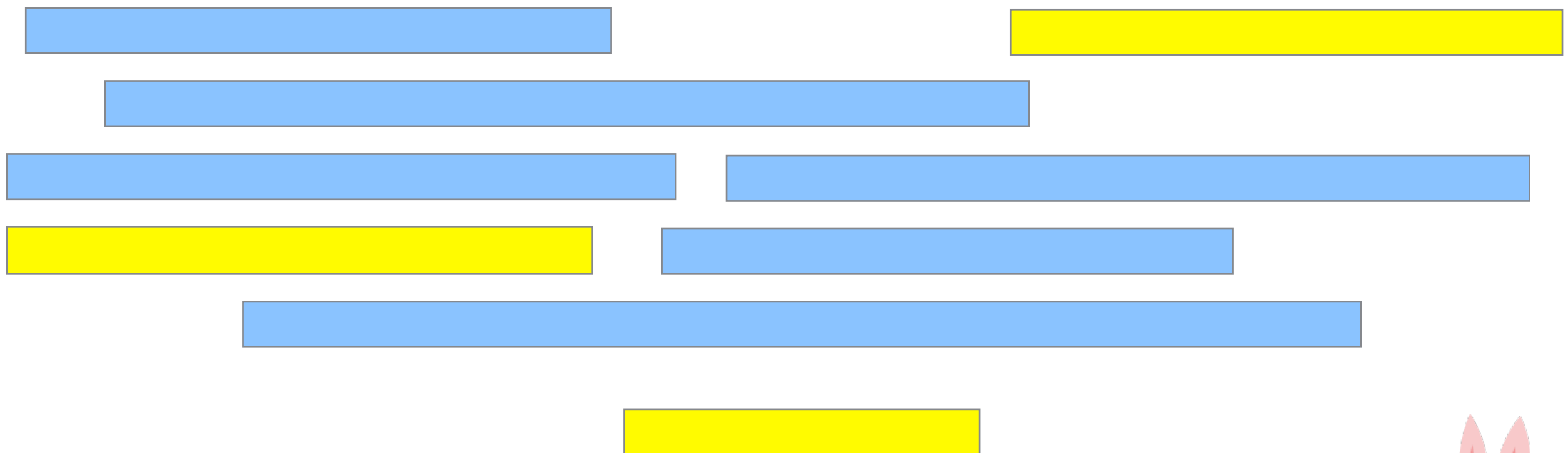
Interval Scheduling

- N개의 구간 중 겹치지 않는 구간을 최대한 많이 선택하라



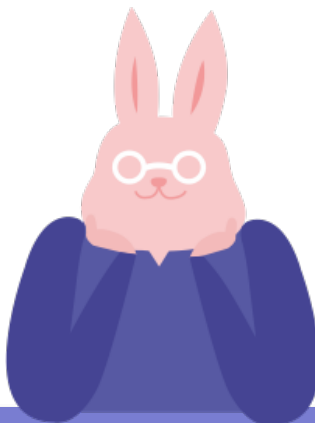
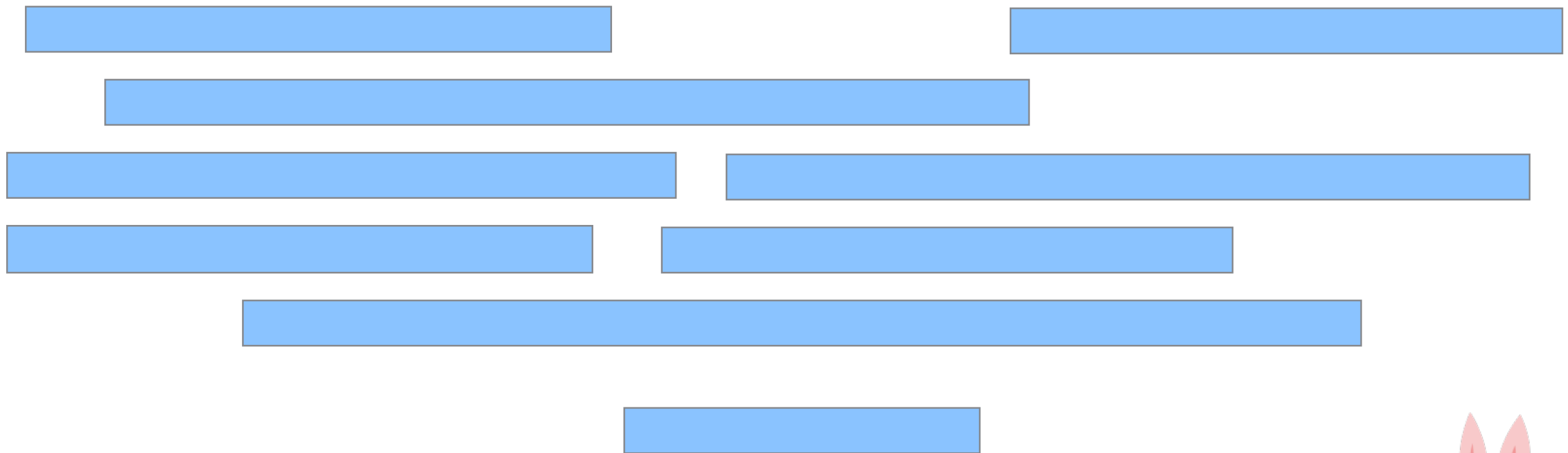
Interval Scheduling

- N개의 구간 중 겹치지 않는 구간을 최대한 많이 선택하라



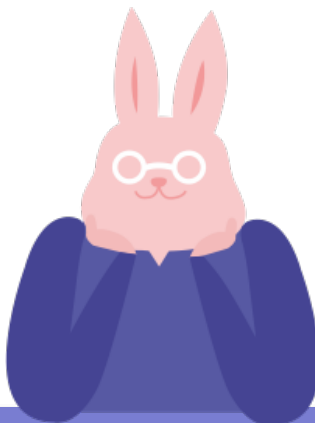
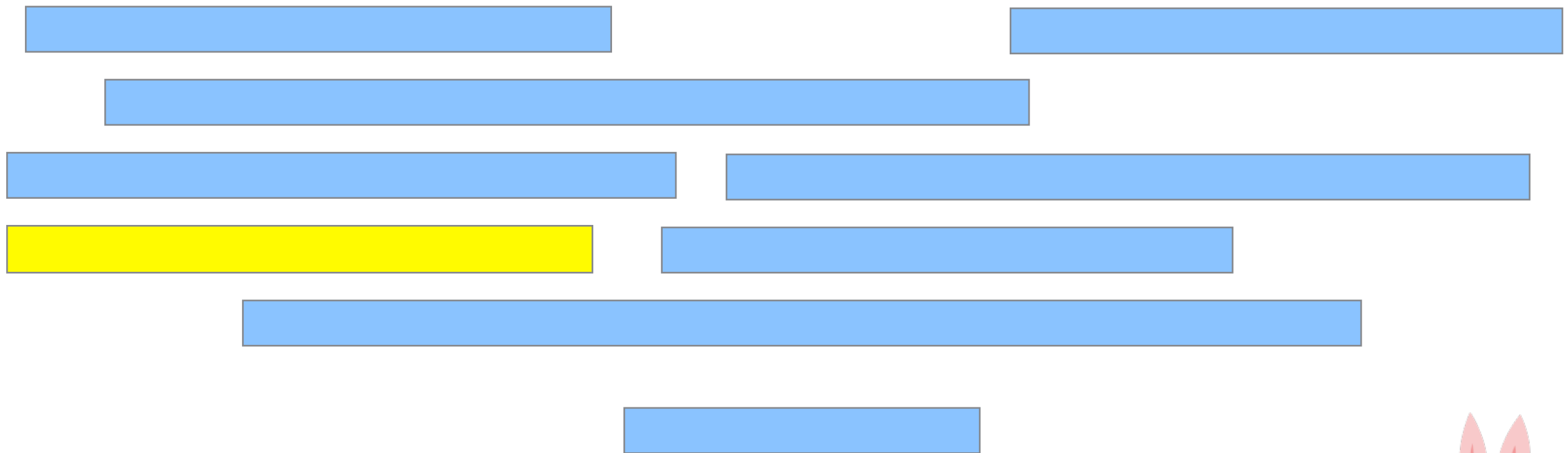
Interval Scheduling

- N개의 구간 중 겹치지 않는 구간을 최대한 많이 선택하라
 - 왼쪽부터 차례대로 선택한다고 하면, 누가 가장 처음이어야 하나 ?



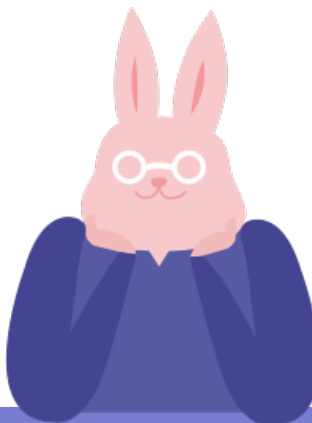
Interval Scheduling

- N개의 구간 중 겹치지 않는 구간을 최대한 많이 선택하라
 - 왼쪽부터 차례대로 선택한다고 하면, **빨리 끝나는 구간이 무조건 좋음!**



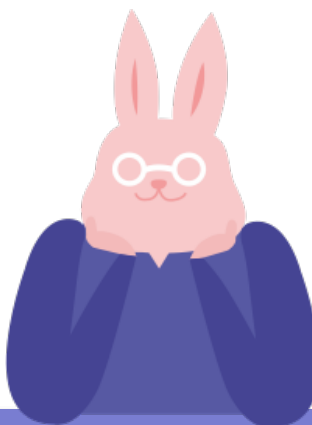
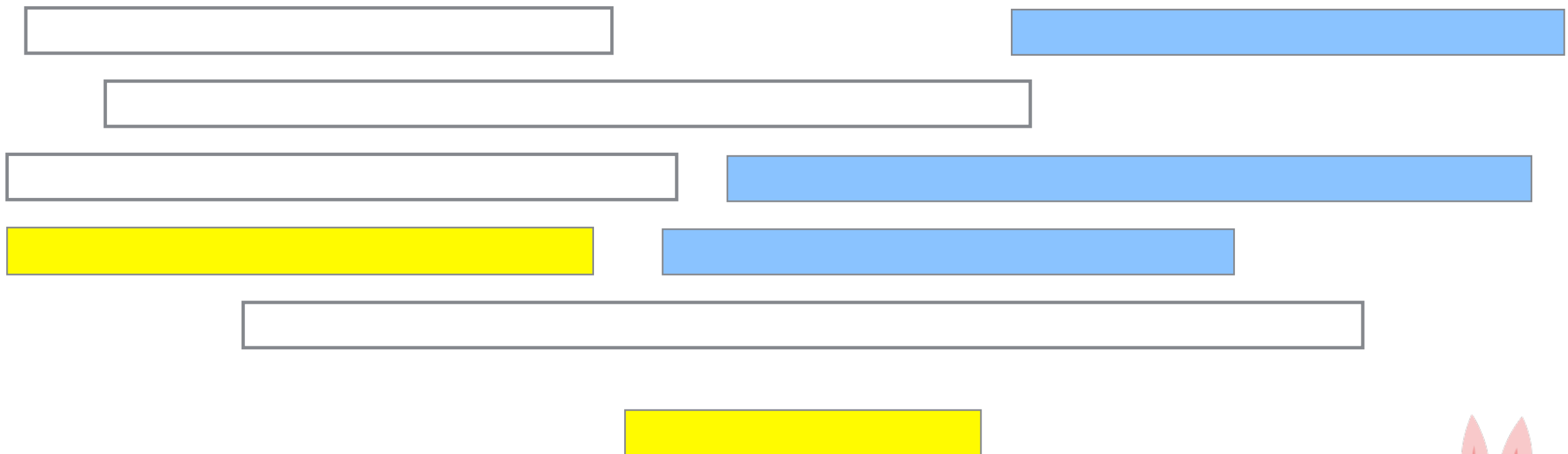
Interval Scheduling

- N개의 구간 중 겹치지 않는 구간을 최대한 많이 선택하라
 - 왼쪽부터 차례대로 선택한다고 하면, **빨리 끝나는 구간이 무조건 좋음!**



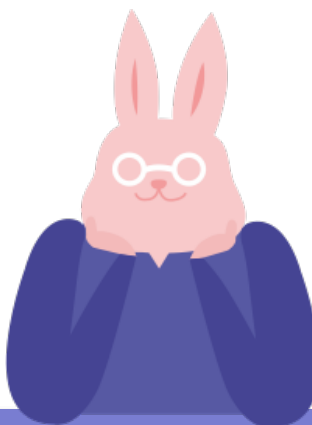
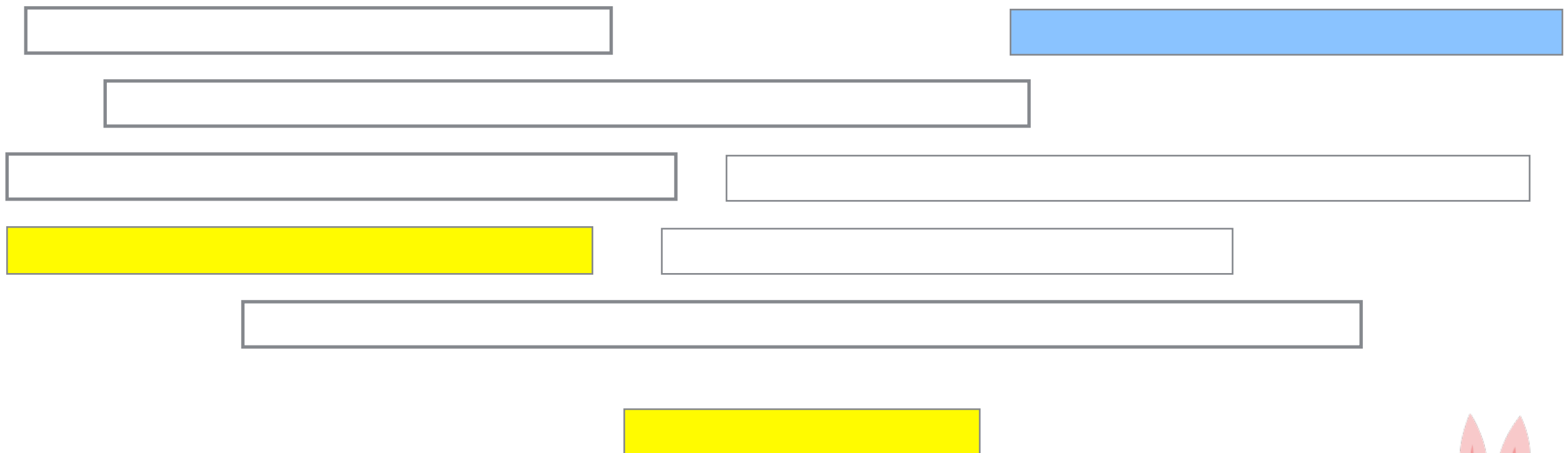
Interval Scheduling

- N개의 구간 중 겹치지 않는 구간을 최대한 많이 선택하라
 - 왼쪽부터 차례대로 선택한다고 하면, **빨리 끝나는 구간이 무조건 좋음!**



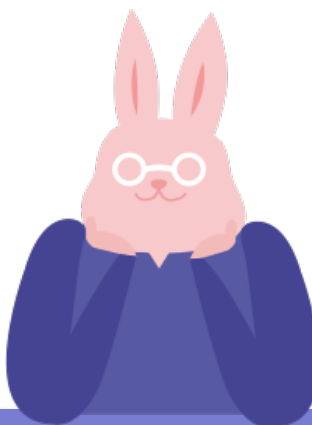
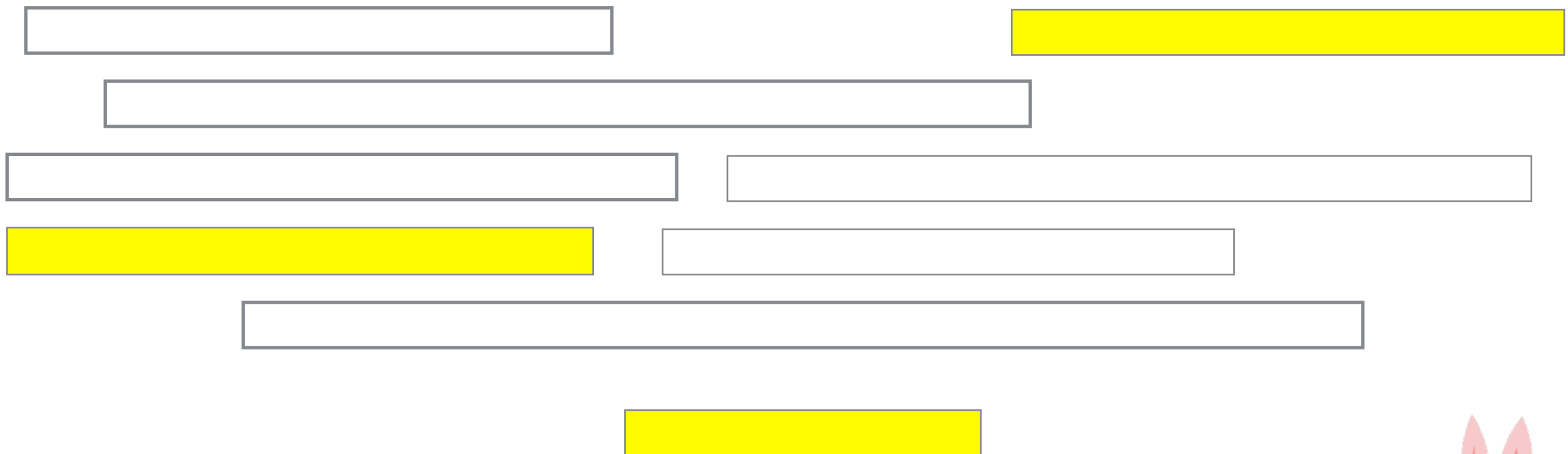
Interval Scheduling

- N개의 구간 중 겹치지 않는 구간을 최대한 많이 선택하라
 - 왼쪽부터 차례대로 선택한다고 하면, **빨리 끝나는 구간이 무조건 좋음!**



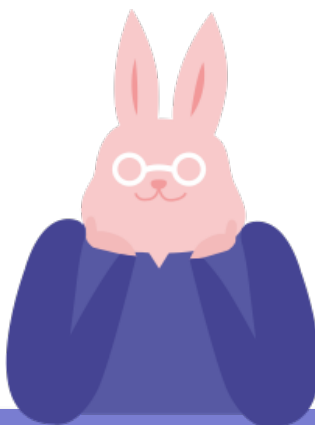
Interval Scheduling

- N개의 구간 중 겹치지 않는 구간을 최대한 많이 선택하라
 - 왼쪽부터 차례대로 선택한다고 하면, **빨리 끝나는 구간이 무조건 좋음!**



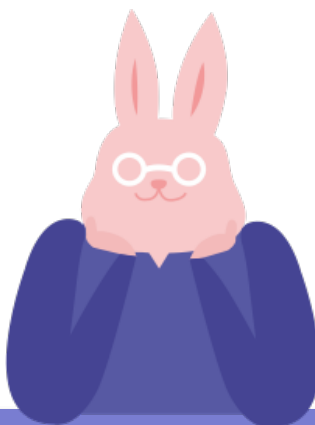
Interval Scheduling

- 시간복잡도
 - 끝나는 시간 순으로 정렬 : $O(n \log n)$



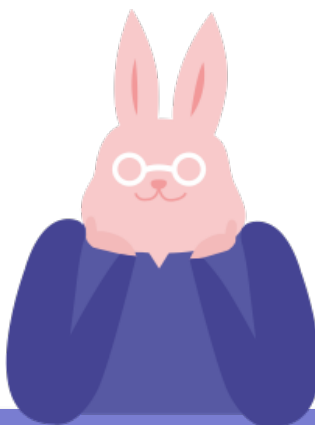
기울기의 최댓값 구하기

- N개의 점 중에서, 두 점을 이었을 때의 기울기의 최댓값은 ?



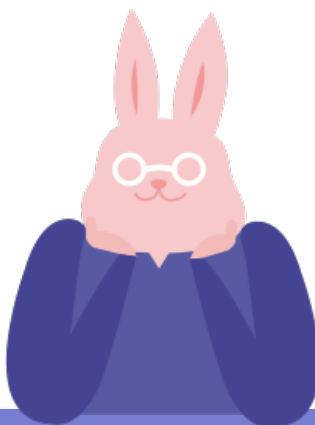
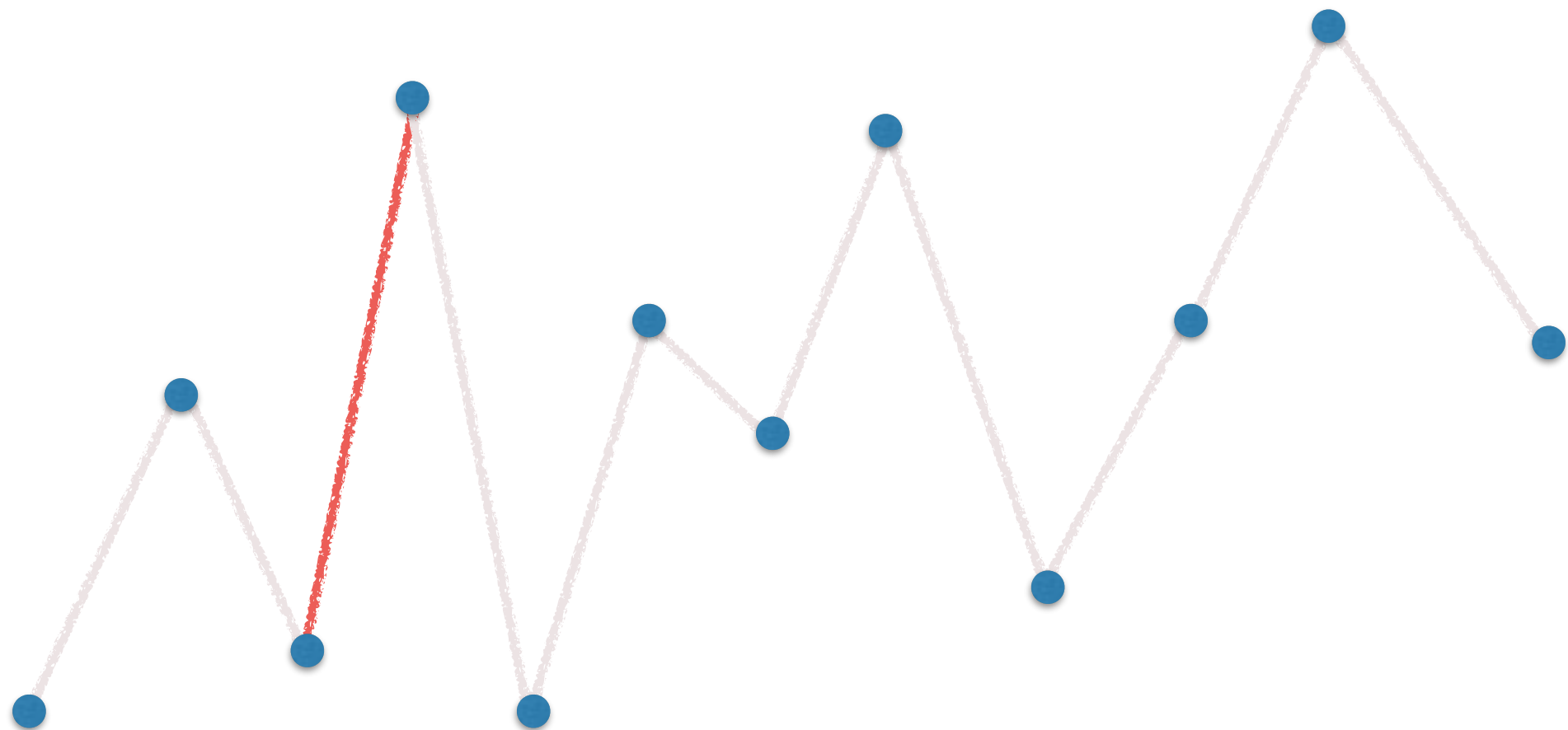
기울기의 최댓값 구하기

- N개의 점 중에서, 두 점을 이었을 때의 기울기의 최댓값은 ?



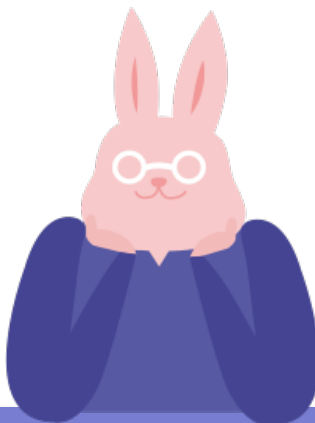
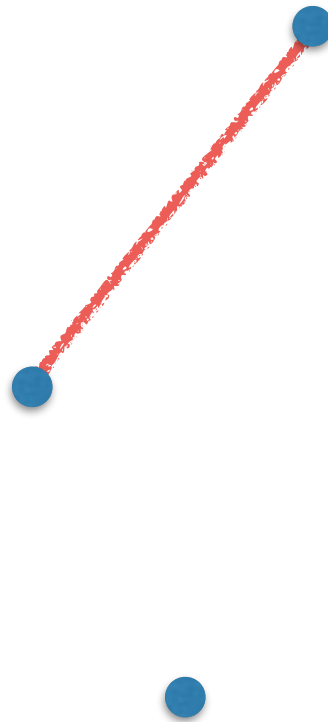
기울기의 최댓값 구하기

- x값으로 정렬 후, 인접한 두 점만 보면 충분하다



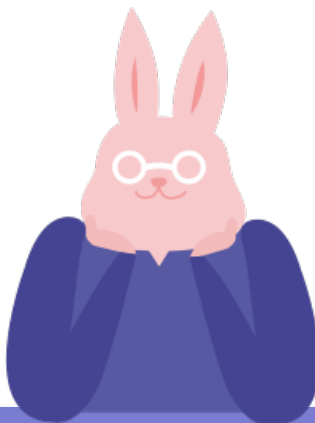
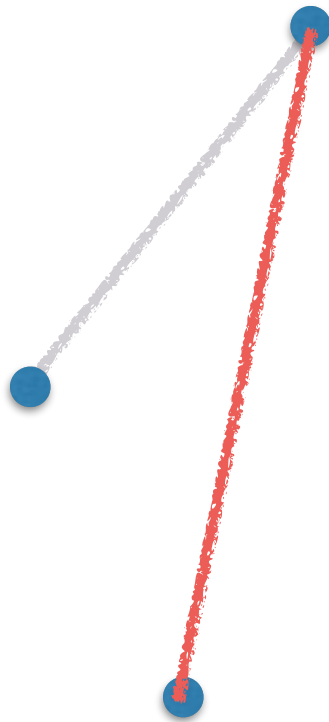
기울기의 최댓값 구하기

- 증명 : 귀류법
 - 인접하지 않은 두 점의 기울기가 최댓값이라 가정하자



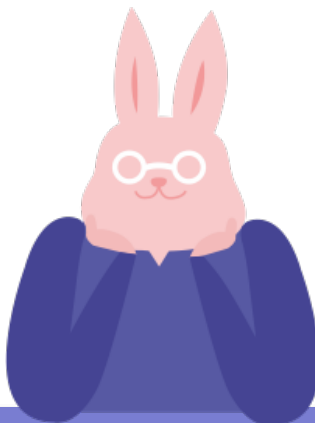
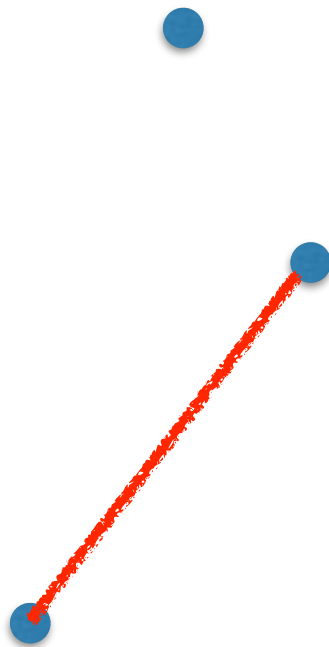
기울기의 최댓값 구하기

- 증명 : 귀류법
 - 인접하지 않은 두 점의 기울기가 최댓값이라 가정하자
 - 그럴 리 없음!



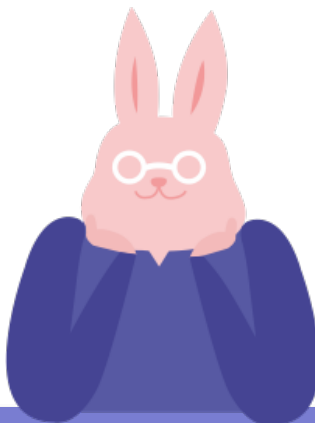
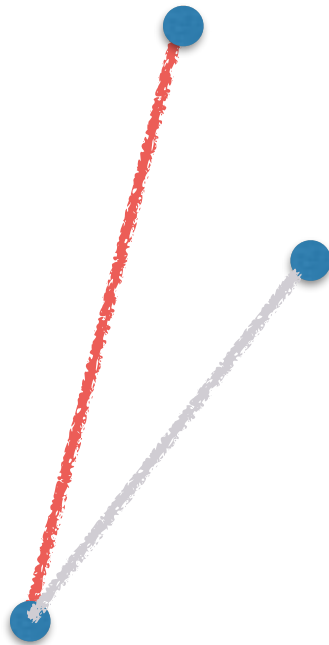
기울기의 최댓값 구하기

- 증명 : 귀류법
 - 인접하지 않은 두 점의 기울기가 최댓값이라 가정하자
 - 그럴 리 없음!



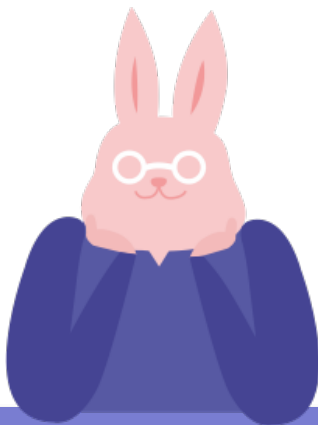
기울기의 최댓값 구하기

- 증명 : 귀류법
 - 인접하지 않은 두 점의 기울기가 최댓값이라 가정하자
 - 그럴 리 없음!



탐욕적 기법 요약

- 단순히 보일 수 있으나, **증명**하는 것이 쉽지 않다
- 대표적 예제를 **증명**까지 꼭 외워두자



감사합니다!

신현규

E-mail : hyungyu.sh@kaist.ac.kr

Kakao : yougatup

