

Data Structure

기말문제풀이종합

<http://smartlead.hallym.ac.kr>

Instructor: Jin Kim
010-6267-8189(033-248-2318)
jinkim@hallym.ac.kr

Office Hours:

선택정렬

여러가지 방법으로 선택정렬을 구현할 수 있다.

기말고사에서는 가장 큰 수를 선택하여 가장 오른쪽 원소와 교환하는 방법을 사용하도록 하자. 가장 작은 수를 선택하여 가장 왼쪽의 원소와 교환하는 방법도 있으나 가장 큰 수를 선택하여 가장 오른쪽 원소와 교환하도록 한다.

2017 문제 1

5, 1, 3, 4, 2, 6을 선택 정렬로 해결하라. 그 과정을 보여라.

◆□답 : 가장 큰수 6을 찾았다.



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

가장 오른쪽의 원소와 교환. 이 경우 자신과 교환하는 셈.



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

2017 문제 1

5, 1, 3, 4, 2, 6을 선택 정렬로 해결하라. 그 과정을 보여라.

◆□ 답 : 빨간색은 위치가 결정된 원소. 빨강의 제외한 원소중 가장 큰 수는 5. ↓

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

가장 오른쪽의 원소와 교환. 이 경우 2와 교환하는 셈.

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

2017 문제 1

5, 1, 3, 4, 2, 6을 선택 정렬로 해결하라. 그 과정을 보여라.

◆□ 답 : 빨간색은 위치가 결정된 원소. 빨강의 제외한 원소중 가장 큰 수는 4. ↓

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

가장 오른쪽의 원소와 교환. 이 경우 4와 교환하는 셈.

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

2017 문제 1

5, 1, 3, 4, 2, 6을 선택 정렬로 해결하라. 그 과정을 보여라.

◆□답 : 계속 같은 과정을 반복. 나는 여기서 중지하였으나, 학생들은 원소하나가 남을때까지 같은 과정을 반복해야 함. 반복하지 않으면 감점.

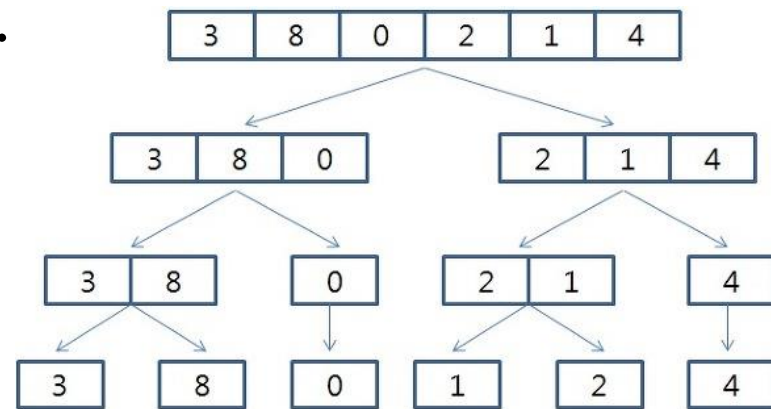
2017 문제 2

제자리 정렬 알고리즘을 설명하라. 제자리 정렬 알고리즘이 아닌 정렬 알고리즘을 하나 적어라.

◆□답:

-제자리 정렬 알고리즘 : 추가적인 메모리공간을 필요로 하지 않는 정렬 알고리즘

-제자리 정렬 알고리즘이 아닌 정렬 알고리즘 : 병합 정렬(Merge sort)
병합 정렬을 같은 크기의 임시 배열을 필요로 한다.



2017 문제 3

안정적인 정렬 알고리즘을 설명하라.

동일한 키가 여러 개 존재할 경우 원래의 키 순으로 정렬하는 알고리즘

버블 정렬, 삽입정렬, 선택정렬

bubble : 인접한 키 값 비교 후 위치를 바꿔주는 방식

insertion : 두번째 인덱스 키 값부터 차례로 앞의 키 값과 비교하여 삽입해주는 방식

Selection : 키 값 중 최대값 혹은 최소값의 위치를 찾아줘 고정해주는 방식

2017 문제 4

$O(n \log n)$, $O(n^2)$ 계열을 정렬 알고리즘이 존재하는 이유를 설명하라.

수백개 이하의 원소 정렬에는 $O(n)$ 알고리즘이 빠르면 그 이상의 원소에는 $O(n \log n)$ 알고리즘이 빠르기 때문이다.

2020 문제 1

keys={8, 2, 4, 6, 9, 7}

선택정렬을 사용하여 정렬하는 과정을 보여라.

이전 문제와 동일하게 해결하면 됨.

2020 문제 2

keys={8, 2, 4, 6, 9, 7}

버블정렬을 사용하여 정렬하는 과정을 보여라.

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

8	2	4	6	9	7
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	8	4	6	9	7
----------	----------	----------	----------	----------	----------

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	8	4	6	9	7
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	8	6	9	7
----------	----------	----------	----------	----------	----------

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	8	6	9	7
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	9	7
----------	----------	----------	----------	----------	----------

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	9	7
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	9	7
----------	----------	----------	----------	----------	----------

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	9	7
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	7	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	7	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	7	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	7	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	7	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	7	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	7	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	8	7	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

2. 버블정렬을 사용하여 정렬하는 과정을 보여라.

Keys={8, 2, 4, 6, 9, 7}

2	4	6	7	8	9
---	---	---	---	---	---

Keys = {8, 2, 4, 6, 9, 7} 버블정렬

8	2	4	6	9	7
2	8	4	6	9	7
2	4	8	6	9	7
2	4	6	8	9	7
2	4	6	8	7	9
2	4	6	8	7	9
2	4	6	7	8	9

$O(n^2)$ 정렬 알고리즘

#서로 인접한 두 원소를 검사하여 정렬하는 알고리즘
-인접한 2개의 레코드를 비교하여 크기가 순서대로 되어
있지 않으면 서로 교환한다.
#선택 정렬과 기본 개념이 유사하다.

2020 문제 3

퀵정렬 역시 다양한 변종 알고리즘들이 존재한다. 그러나 우리는 다음과 같이 문제를 해결하는 것을 원칙으로 한다.

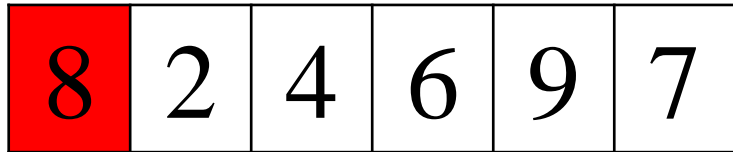
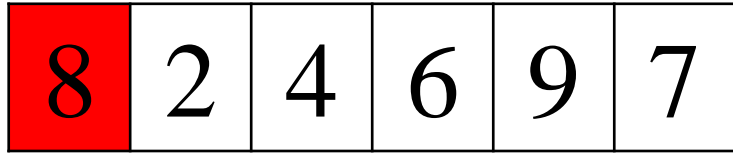
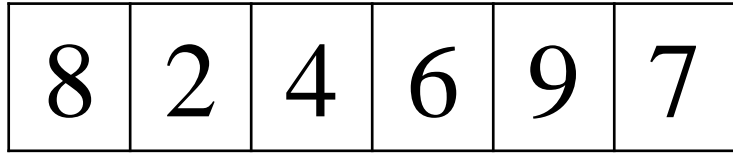
- 1) 제일 왼쪽을 피벗으로 선정한다.
- 2) 왼쪽에서 오른쪽으로 가며 피벗보다 작은수를 찾는다. 찾았으면 제일 왼쪽 원소(피벗을 제외한)와 교환
- 3) 마지막으로 교환한 왼쪽원소와 피벗과 교환한다. 이 결과 피벗보다 작은 것은 피벗왼쪽에, 피벗보다큰 원소는 피벗오른쪽에 위치하게 되며 적어도 피벗은 자기 위치를 찾은 셈이 된다.
- 4) 반드시 위와 같은 방법으로 퀵정렬을 수행해야 정답으로 인정.

2020 문제 3

keys={8, 2, 4, 6, 9, 7}

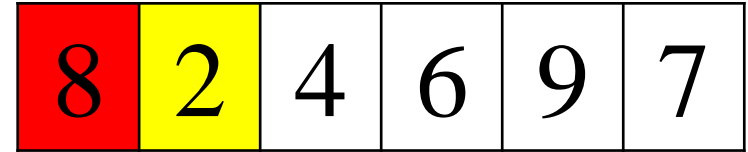
퀵정렬을 사용하여 정렬하는 과정을 보여라.

{8, 2, 4, 6, 9, 7} 의 경우



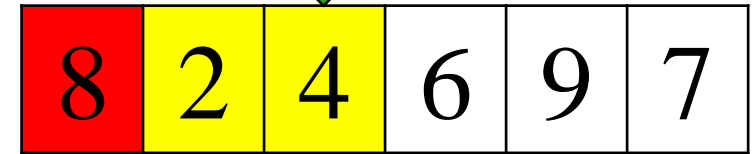
↑ 피벗보다 작은 원소발견

↓ 오른쪽으로 하나 이동후 교환.
이 경우 자기 자신과 교환



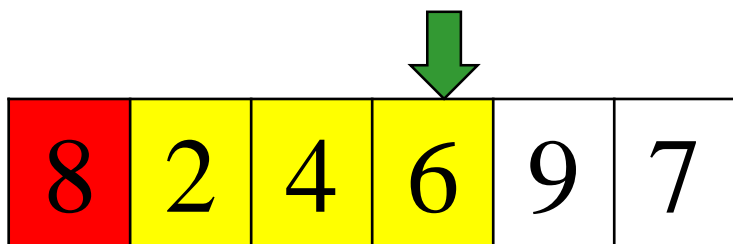
피벗보다 작은 원소발견

오른쪽으로 하나 이동후 교환
이 경우 자기 자신과 교환

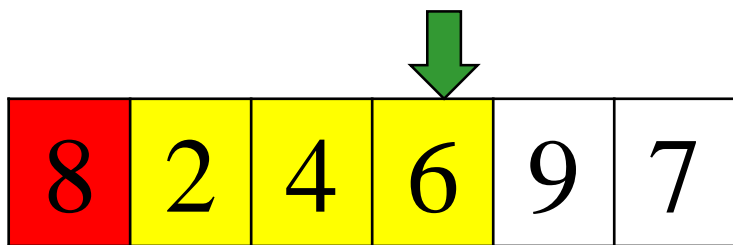


마찬가지 경우



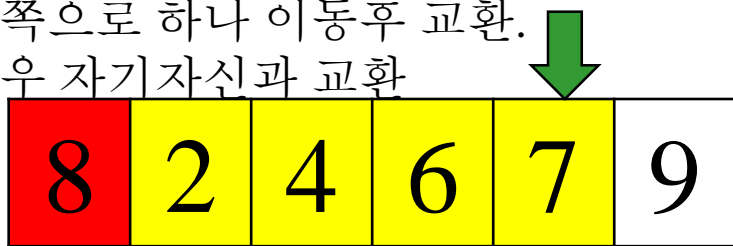


↑ 피봇보다 작은 원소아님

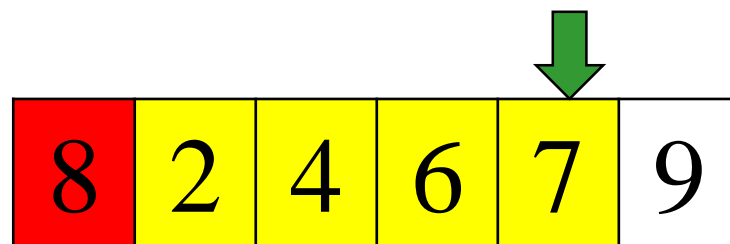


↑ 피봇보다 작은 원소

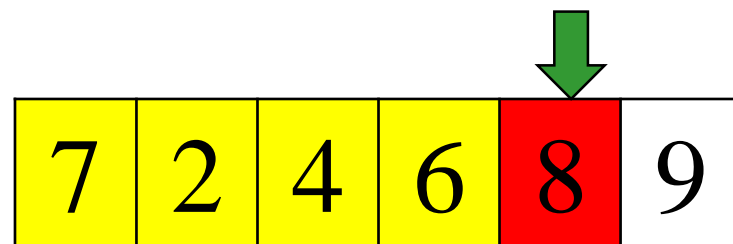
오른쪽으로 하나 이동후 교환.
이경우 자기자신과 교환



↑ 피봇보다 작은 원소



모든원소들을 점검하였으므로 한판의 패스끝



피봇과 가장마지막에 교환한 왼쪽수인 7과 교환
피봇8은 위치가 결정됨. 8의 왼쪽원소들을 퀵정렬.
8의 오른쪽원소들을 퀵정렬수행

7	2	4	6	8	9
---	---	---	---	---	---

7을 다시 피벗으로 하여 {7,2,4,6}에 대해
퀵정렬을 수행하면 아래와 같은 결과

6	2	4	7	8	9
---	---	---	---	---	---

6을 다시 피벗으로 하여 {6, 2, 4}에 대해
퀵정렬을 수행하면 아래와 같은 결과

4	2	6	7	8	9
---	---	---	---	---	---

4를 다시 피벗으로 하여 {4,2}에 대해
퀵정렬을 수행하면 아래와 같은 결과

2	4	6	7	8	9
---	---	---	---	---	---

원소가 하나일 경우 이미 정렬. 아무 작업도 하지 않음
아래 결과

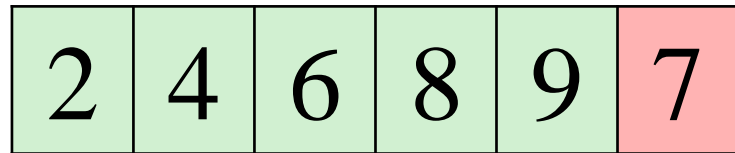
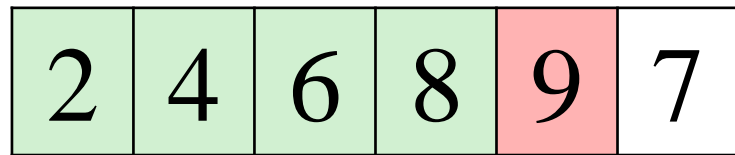
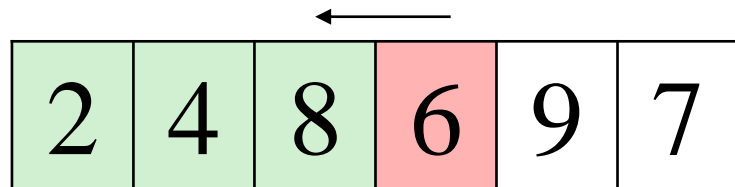
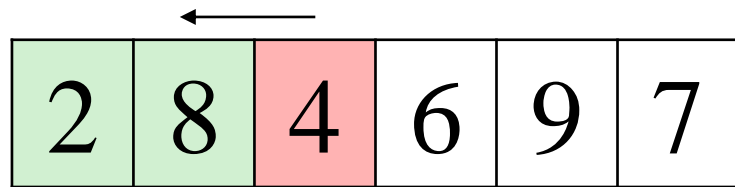
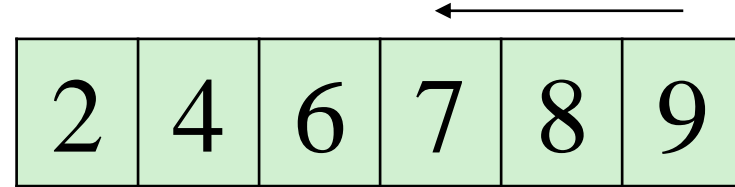
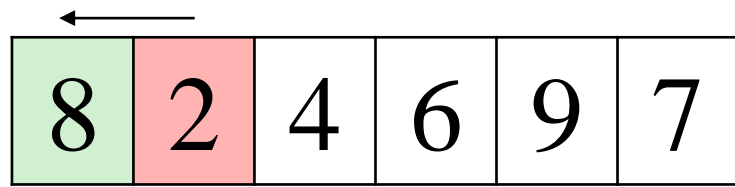
2	4	6	7	8	9
---	---	---	---	---	---

2020 문제 4

keys={8, 2, 4, 6, 9, 7}

삽입정렬을 사용하여 정렬하는 과정을 보여라.

2020 문제 4



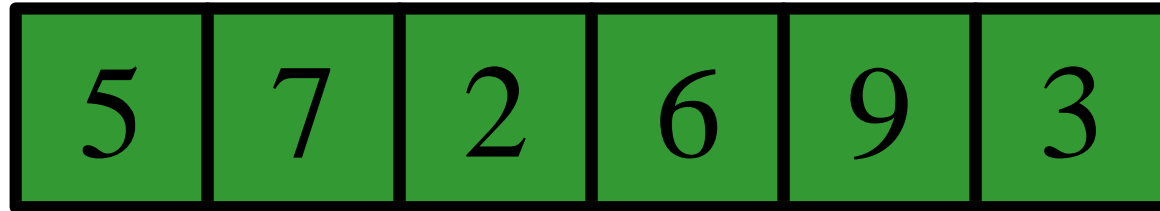
문제 5

문제 : $O(n \log n)$, $O(n^2)$ 계열 정렬 알고리즘을 각각 세 개씩 적어라.
(List three sorting algorithm with each $O(n \log n)$, $O(n^2)$)

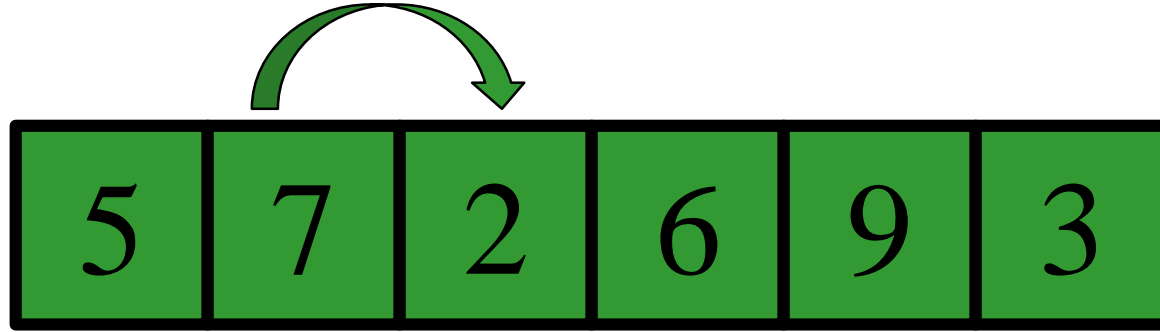
답 : $O(n \log n)$ – 퀵정렬, 힙정렬, 합병정렬
 $O(n^2)$ – 버블정렬, 선택정렬, 삽입정렬

문제 6

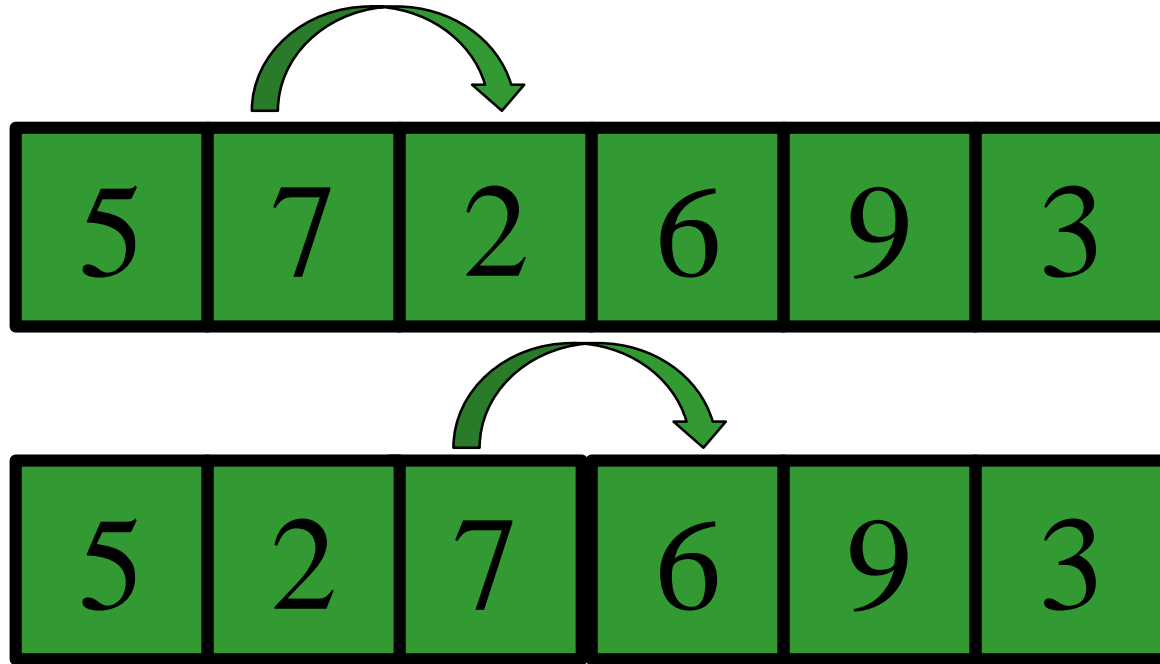
문제 : 5, 7, 2, 6, 9, 3을 버블 정렬로 해결하라. 그 과정을 보여라.
(Sort the elements 5, 7, 2, 6, 9, 3 using bubble sort. Show the detailed process.)



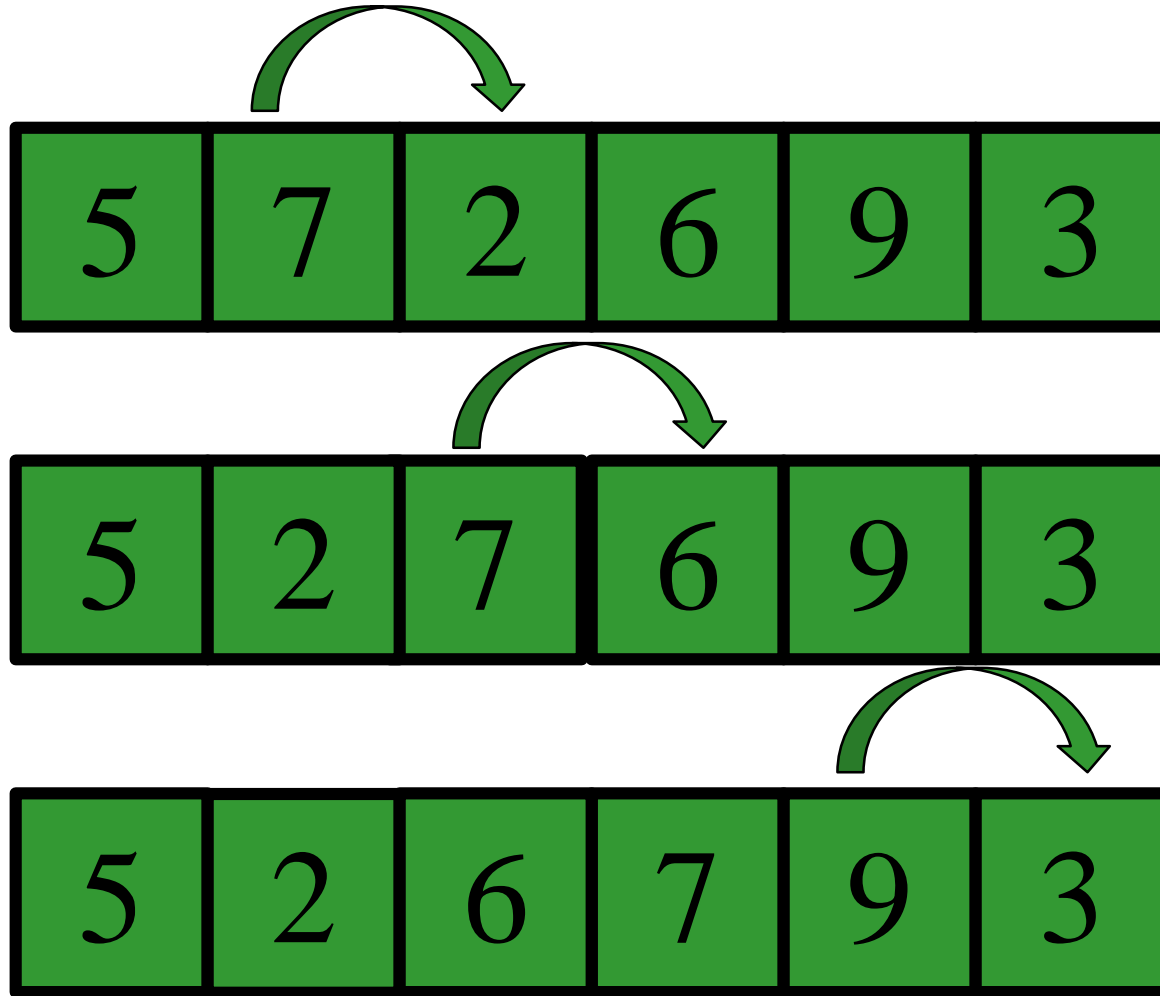
문제 6



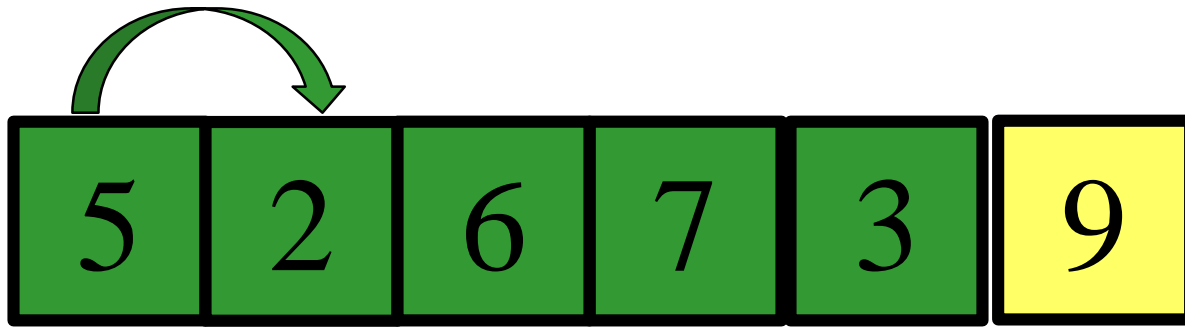
문제 6



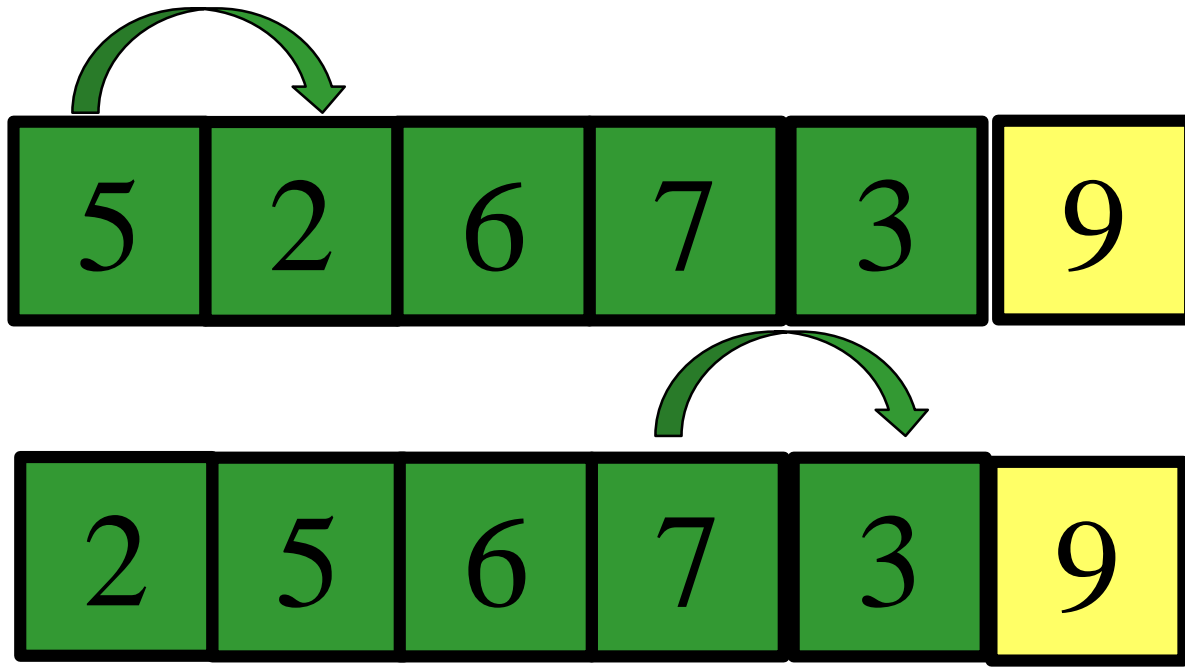
문제 6



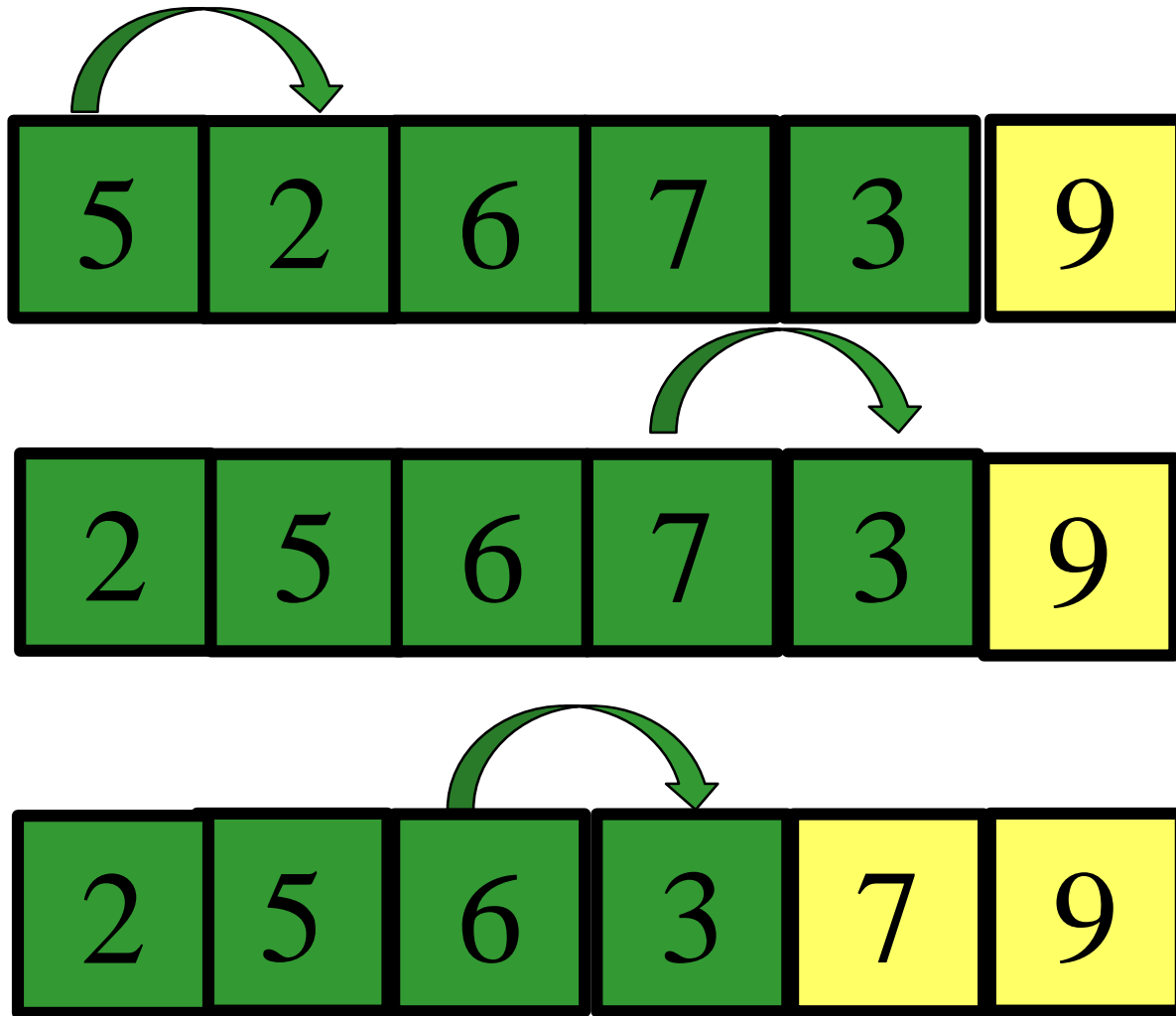
문제 6



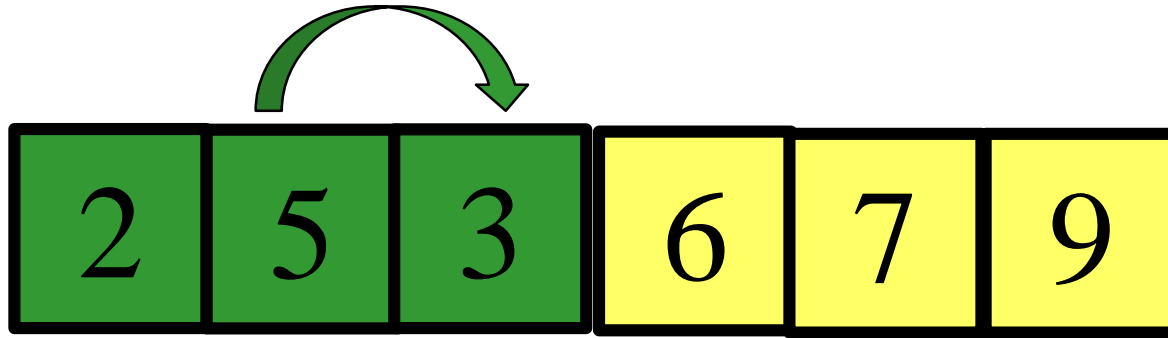
문제 6



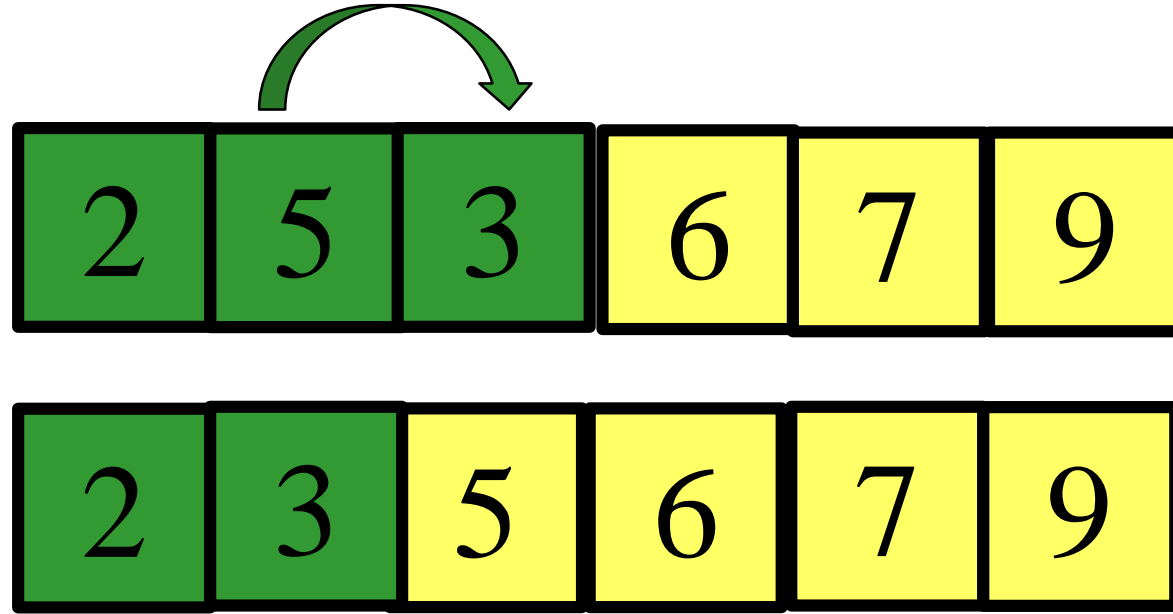
문제 6



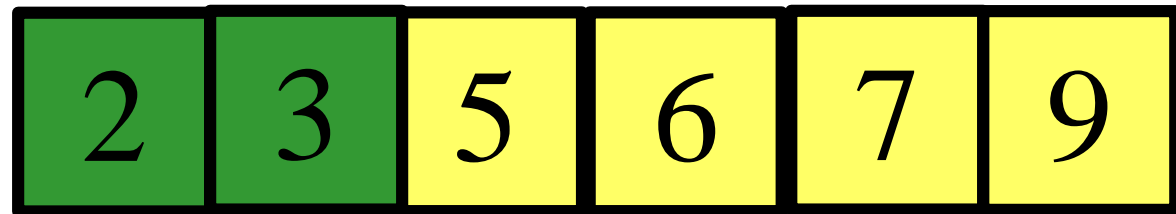
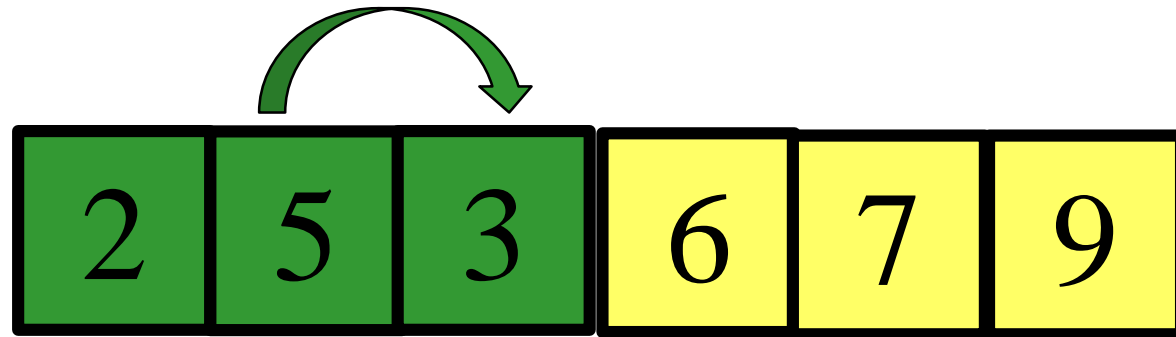
문제 6



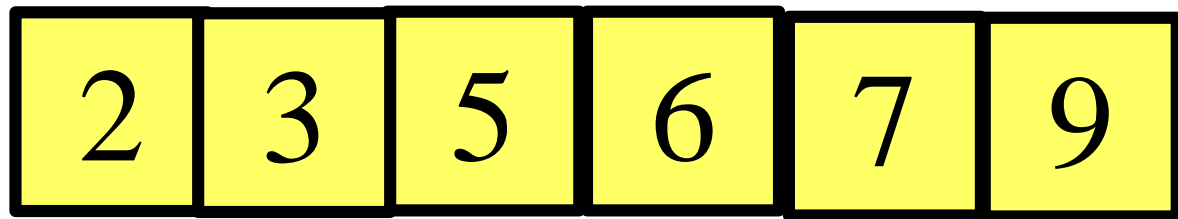
문제 6



문제 6



답 :



문제 7

문제 : . 6, 5, 9, 12, 3, 4를 퀵 정렬로 해결하라. 그 과정을 보여라.

답: 나는 중간과정만을 보일터이니, 학생들은 전과정을 보여야함.

6	5	9	12	3	4
---	---	---	----	---	---

6	5	9	12	3	4
---	---	---	----	---	---

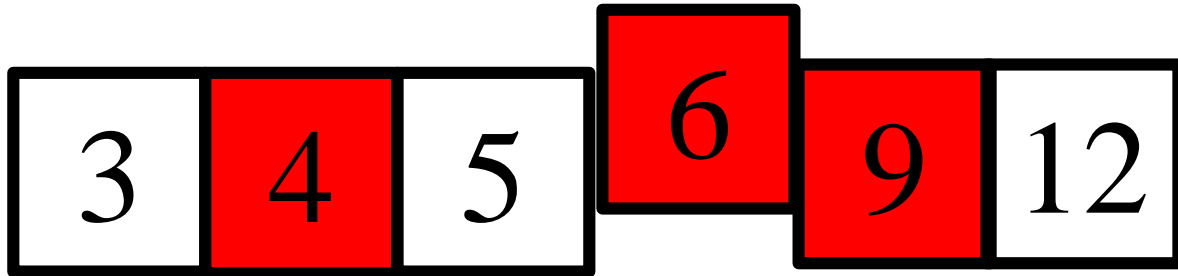
6	5	3	4	9	12
---	---	---	---	---	----

4	5	3	6	9	12
---	---	---	---	---	----

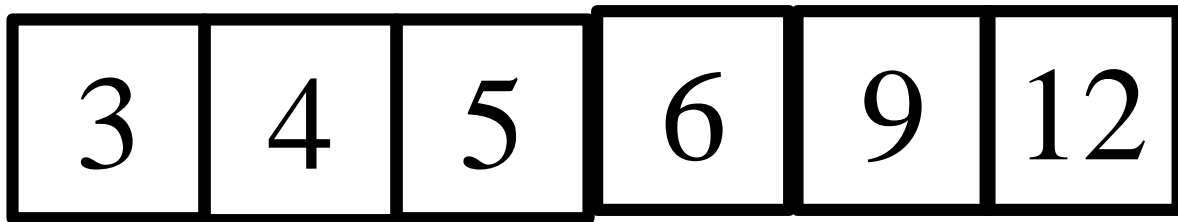
4	5	3	6	9	12
---	---	---	---	---	----

4	3	5	6	9	12
---	---	---	---	---	----

3	4	5	6	9	12
---	---	---	---	---	----



최종



문제 8

문제 : 98, 23, 45, 14, 6, 67, 33, 42를 합병정렬로 해결하라. 그 과정을 보여라. (Sort the elements 98, 23, 45, 14, 6, 67, 33, 42 using merge sort. Show the detailed process.)

98	23	45	14	6	67	33	42
----	----	----	----	---	----	----	----

문제 8

98	23	45	14	6	67	33	42
----	----	----	----	---	----	----	----

문제 8

98	23	45	14	6	67	33	42
----	----	----	----	---	----	----	----

98	23	45	14
----	----	----	----

6	67	33	42
---	----	----	----

문제 8

98	23	45	14	6	67	33	42
----	----	----	----	---	----	----	----

98	23	45	14
----	----	----	----

6	67	33	42
---	----	----	----

98	23
----	----

45	14
----	----

6	67
---	----

33	42
----	----

문제 8

98	23	45	14	6	67	33	42
----	----	----	----	---	----	----	----

98	23	45	14
----	----	----	----

6	67	33	42
---	----	----	----

98	23
----	----

45	14
----	----

6	67
---	----

33	42
----	----

98

23

45

14

6

67

33

42

문제 8

98

23

45

14

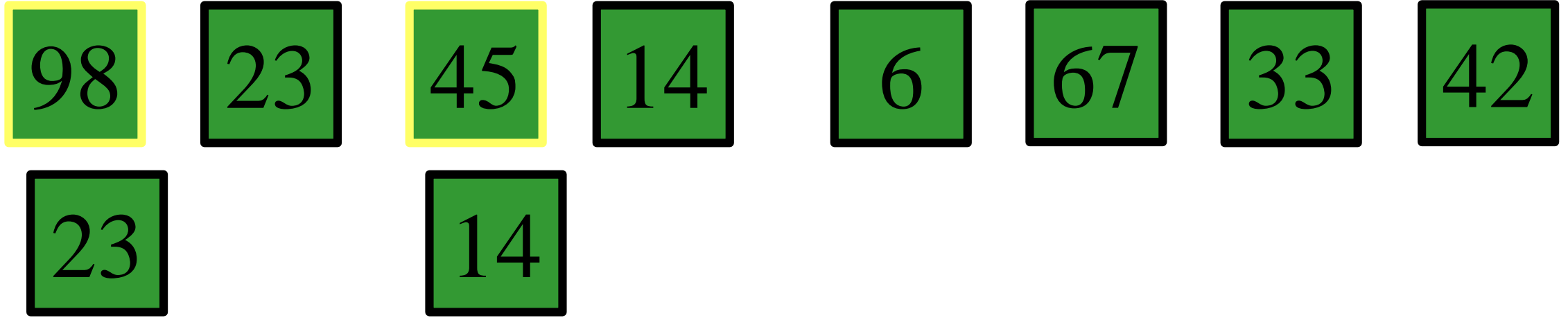
6

67

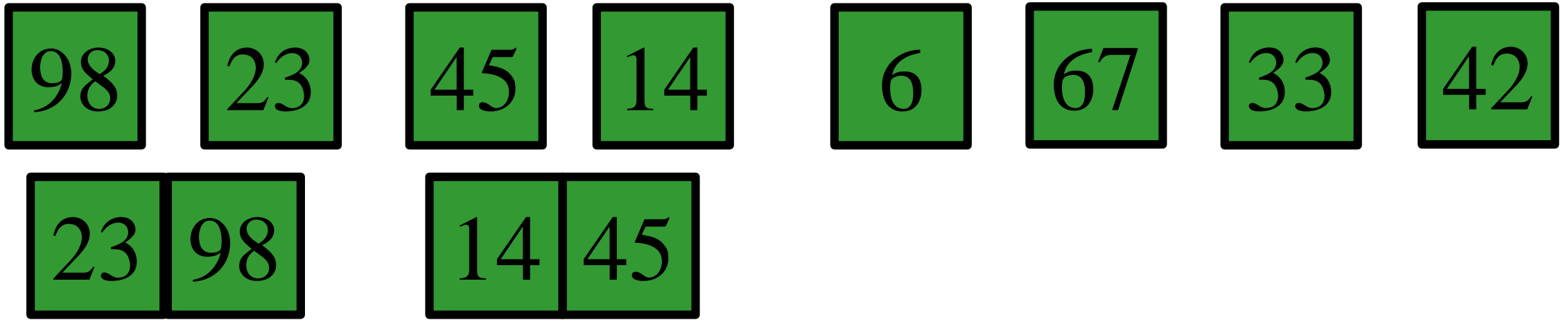
33

42

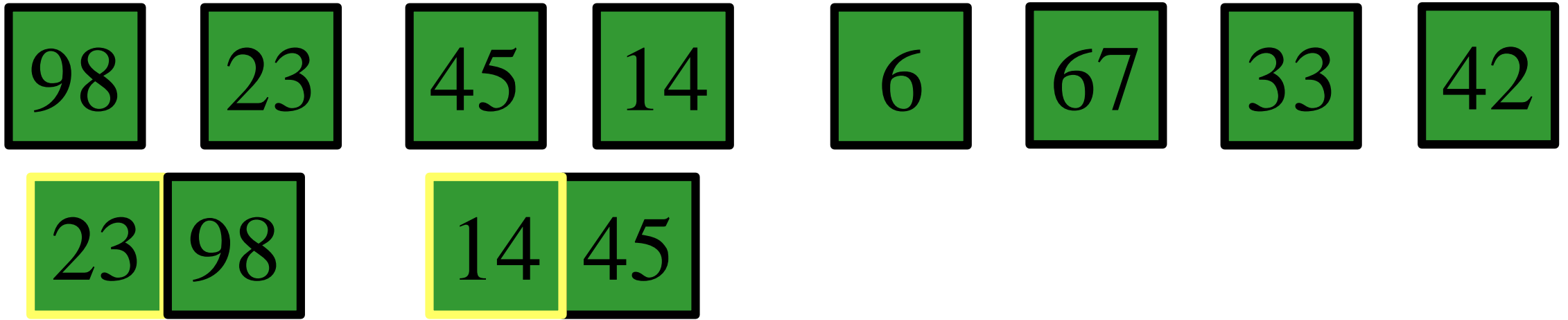
문제 8



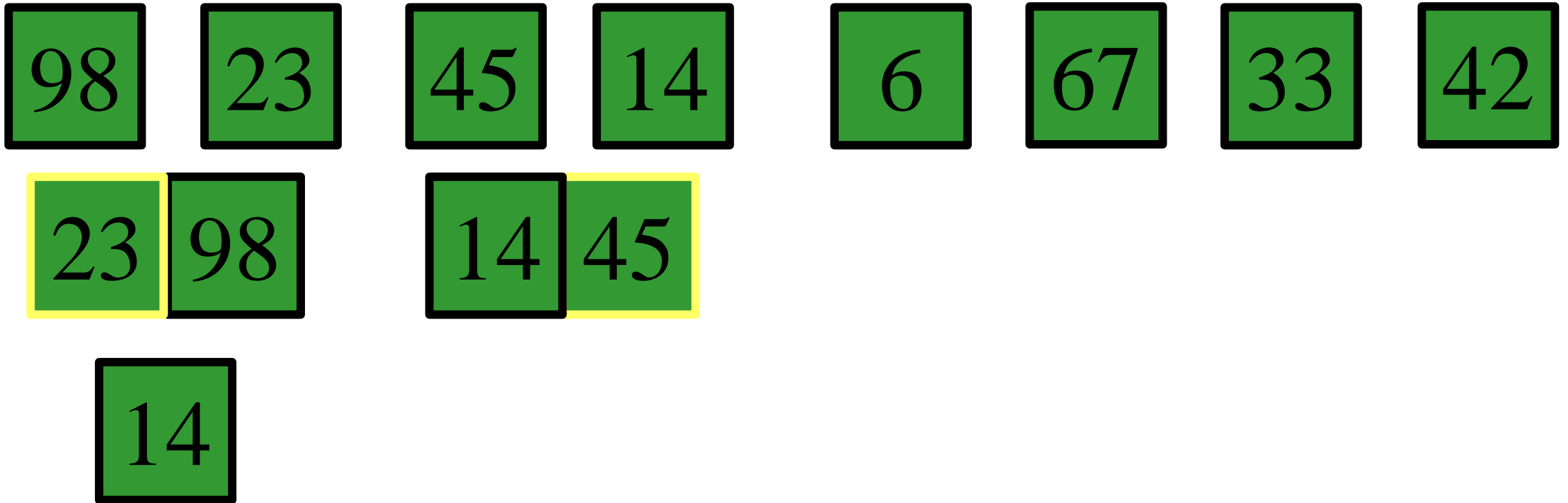
문제 8



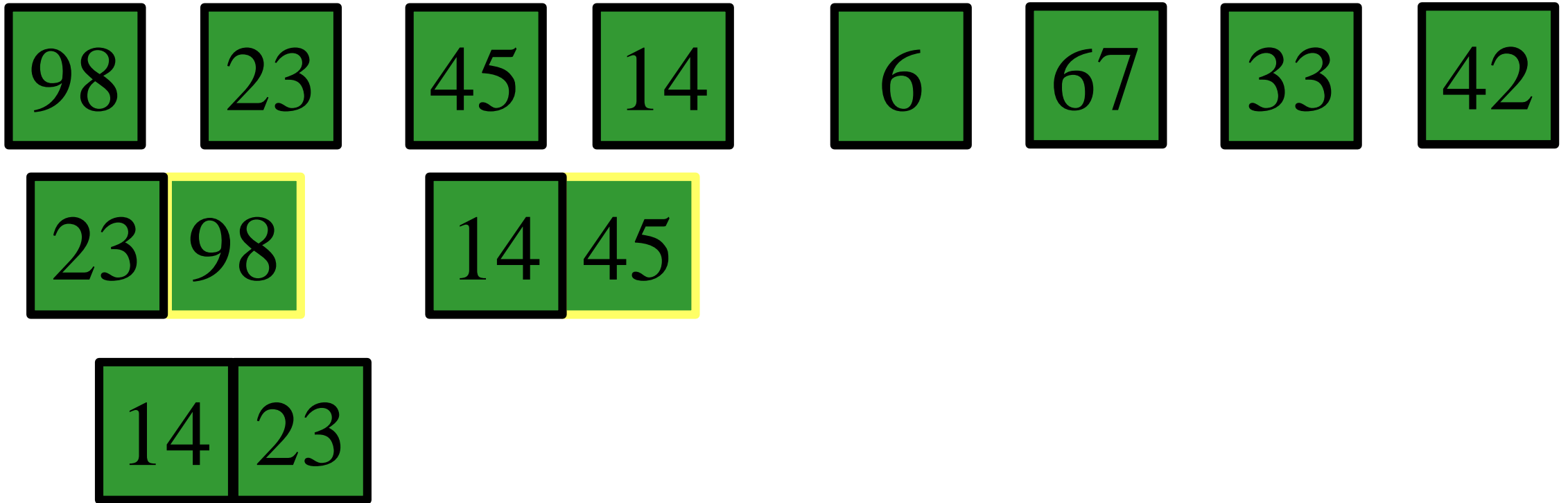
문제 8



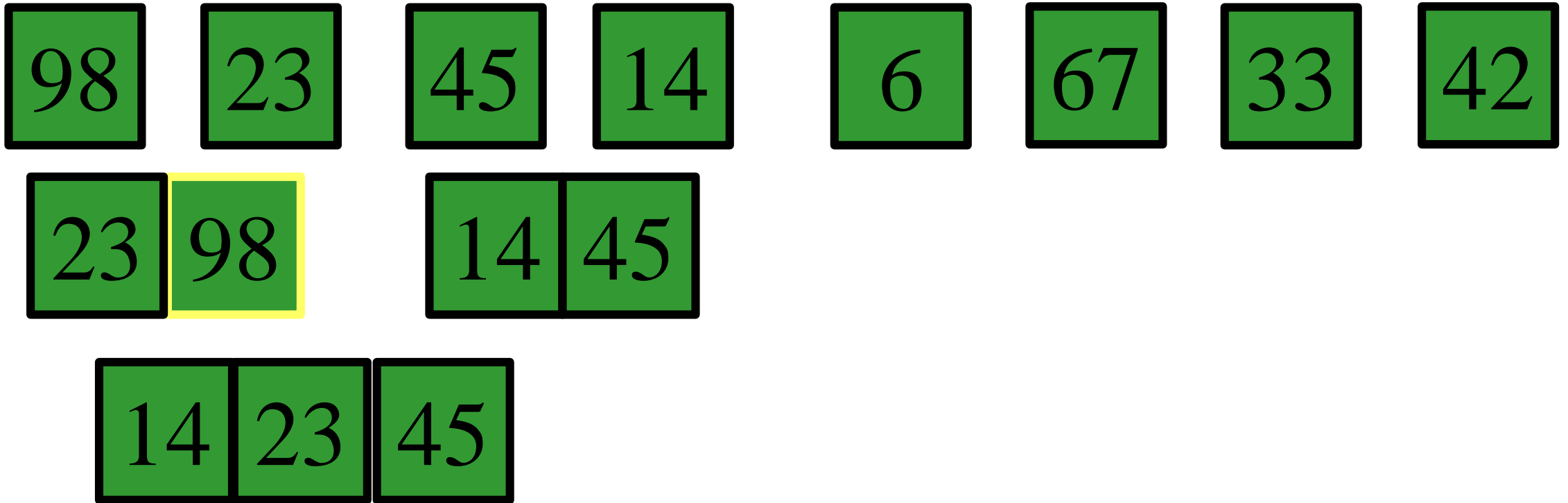
문제 8



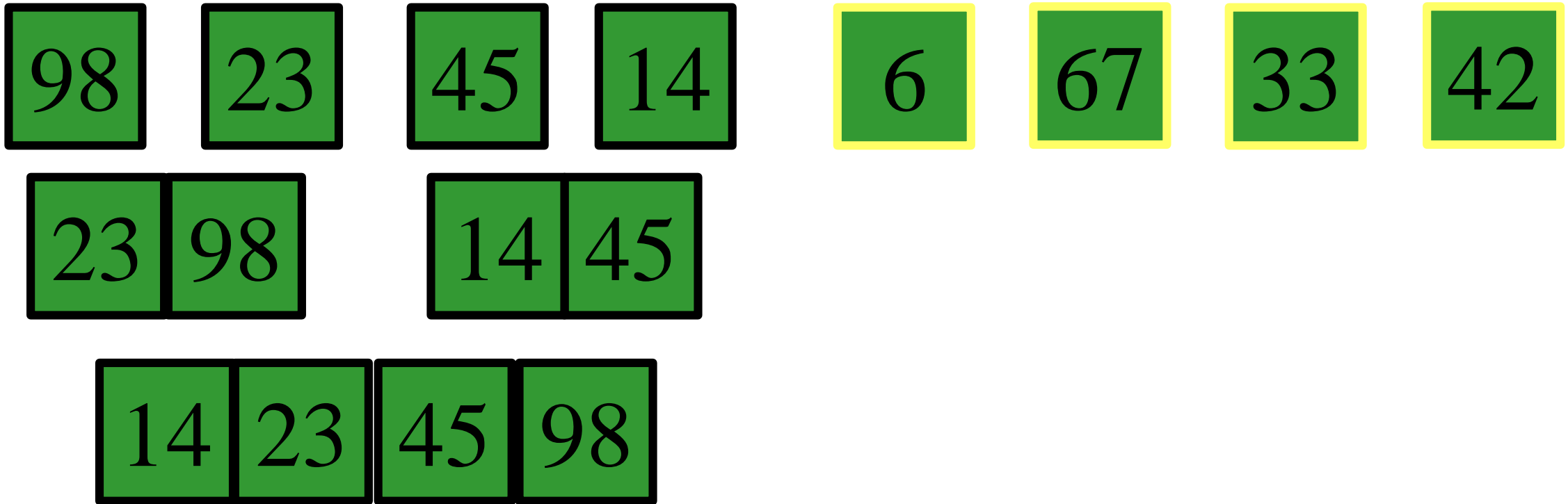
문제 8



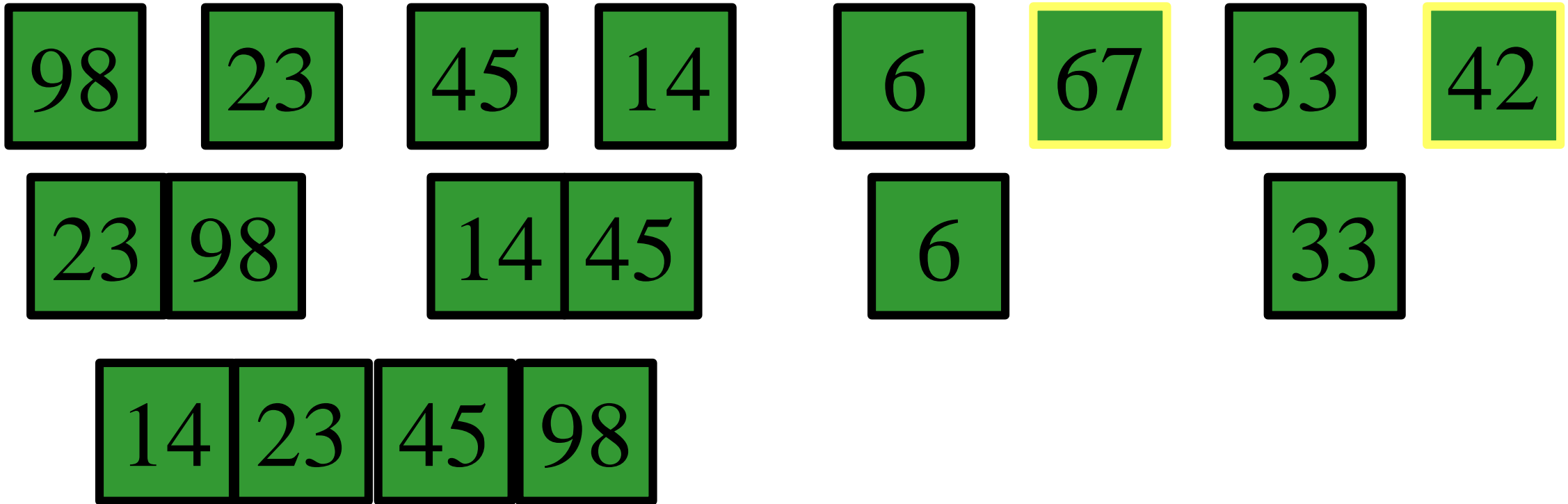
문제 8



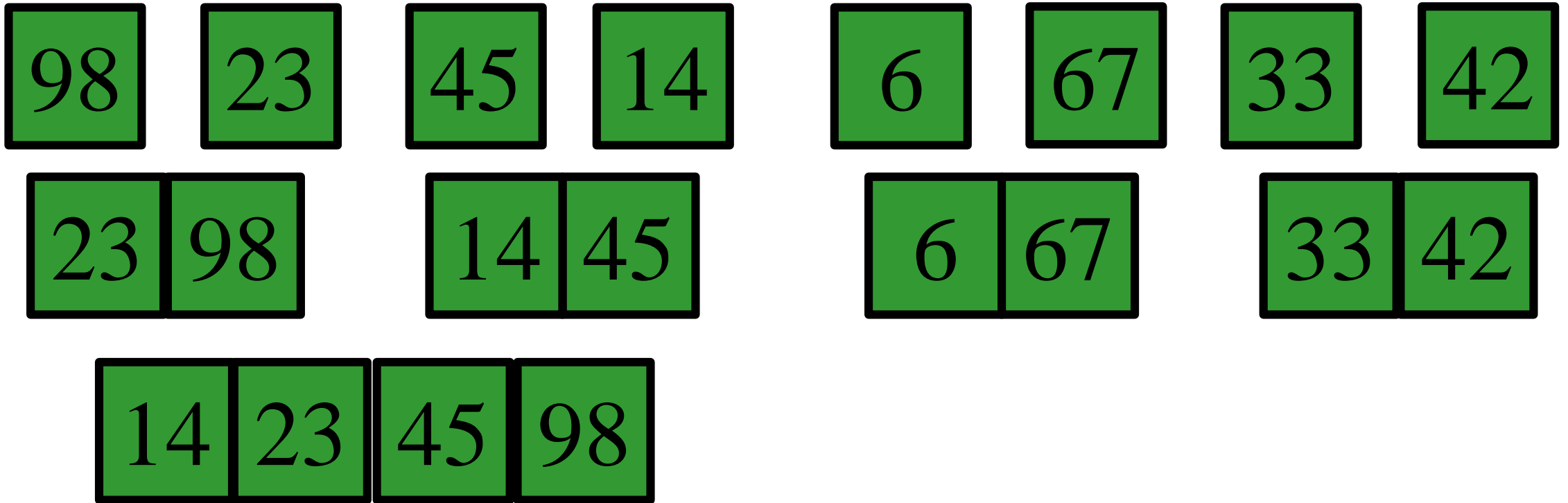
문제 8



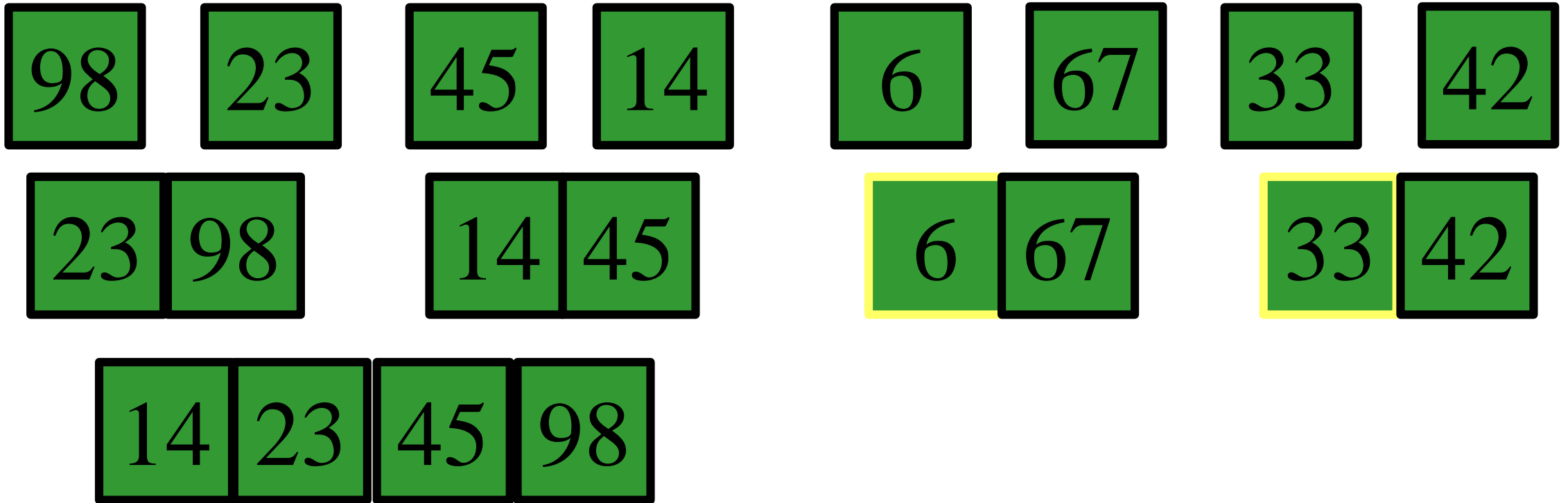
문제 8



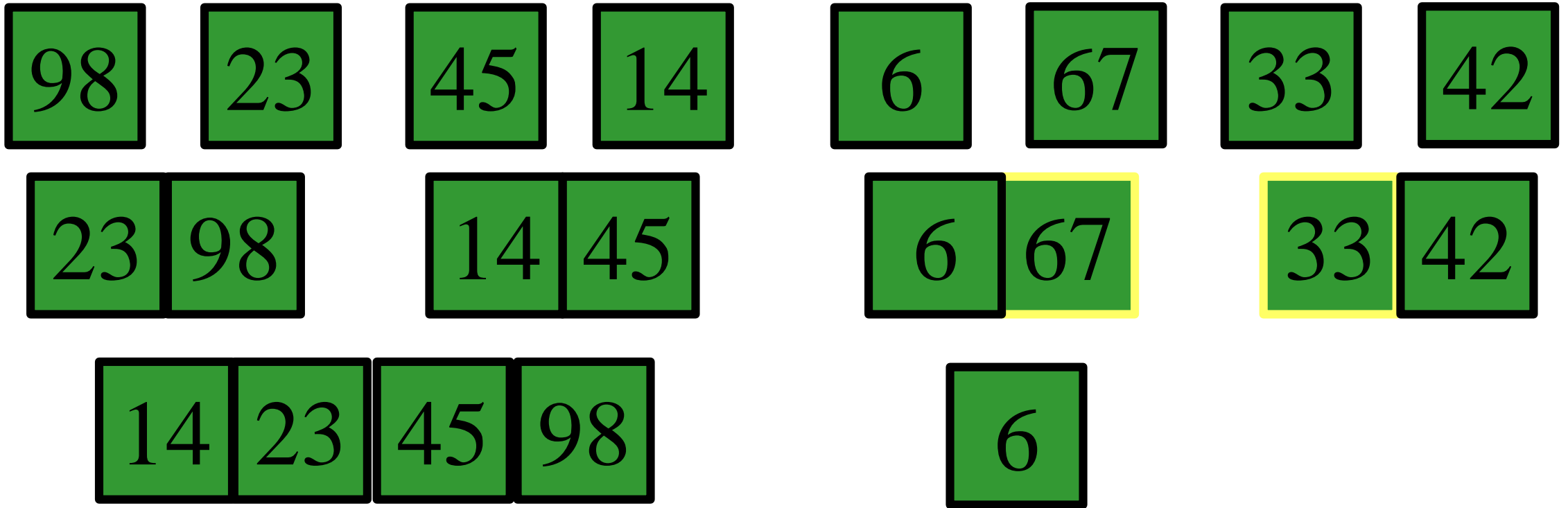
문제 8



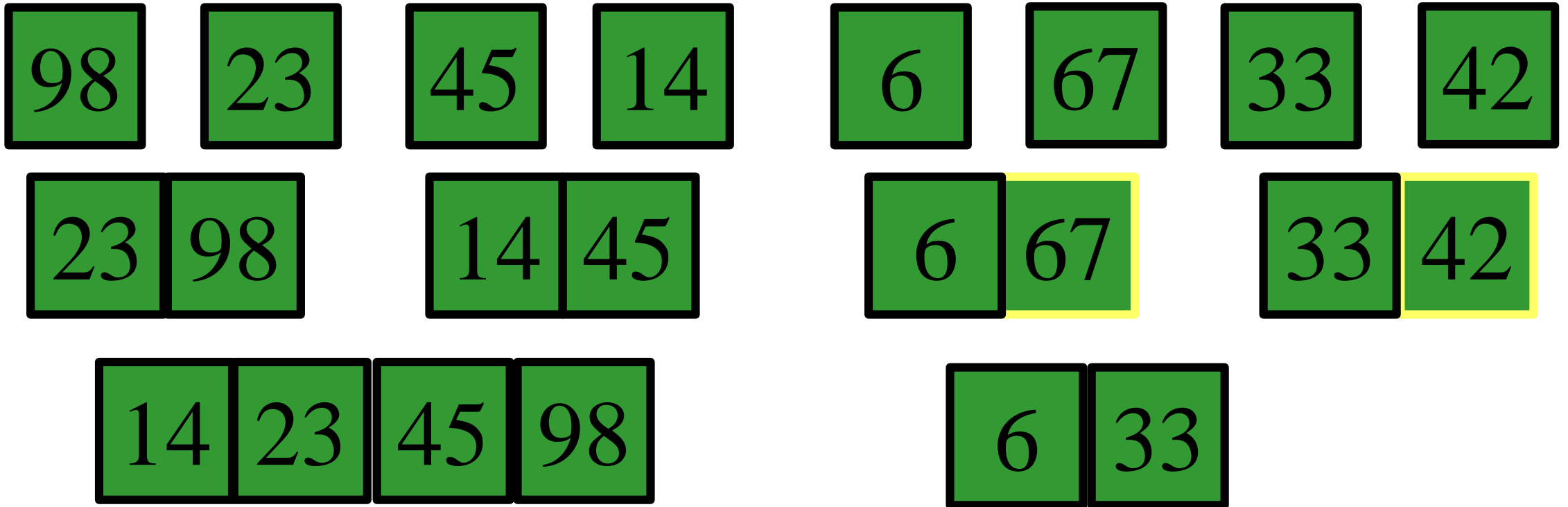
문제 8



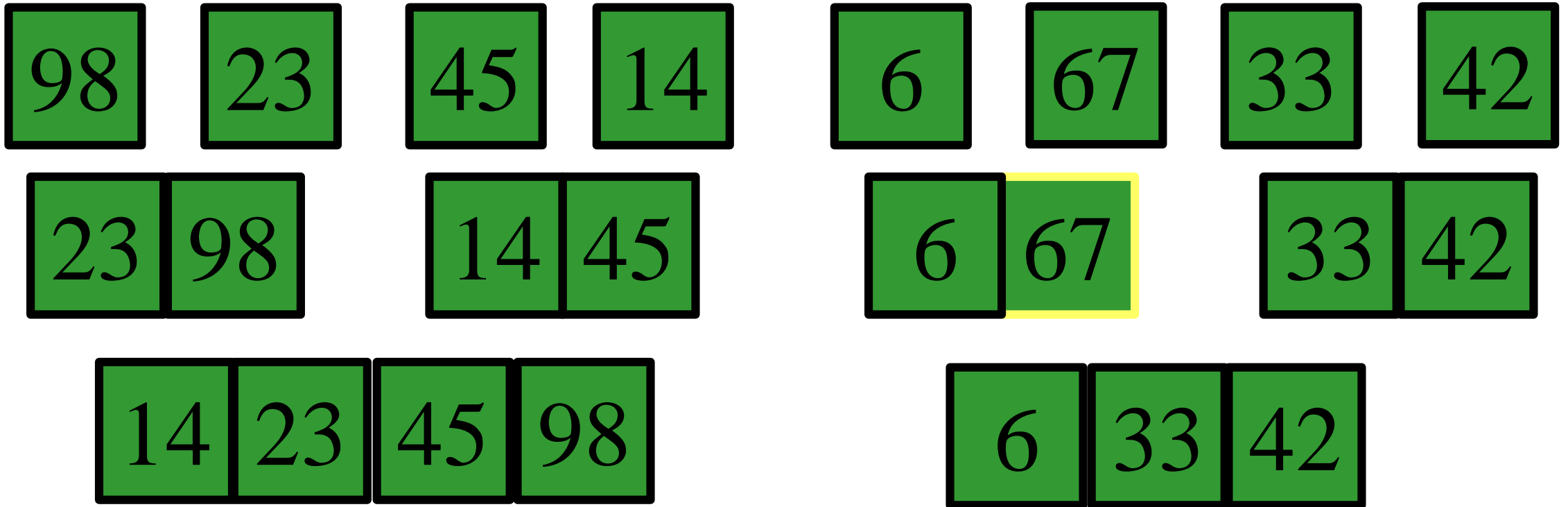
문제 8



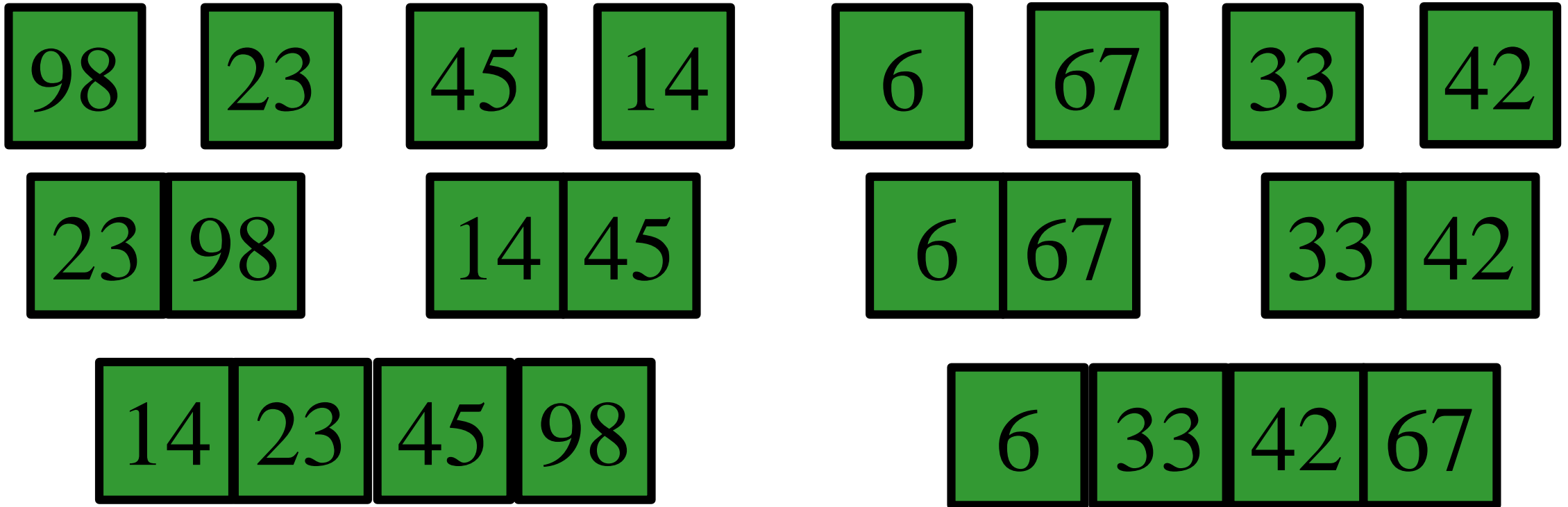
문제 8



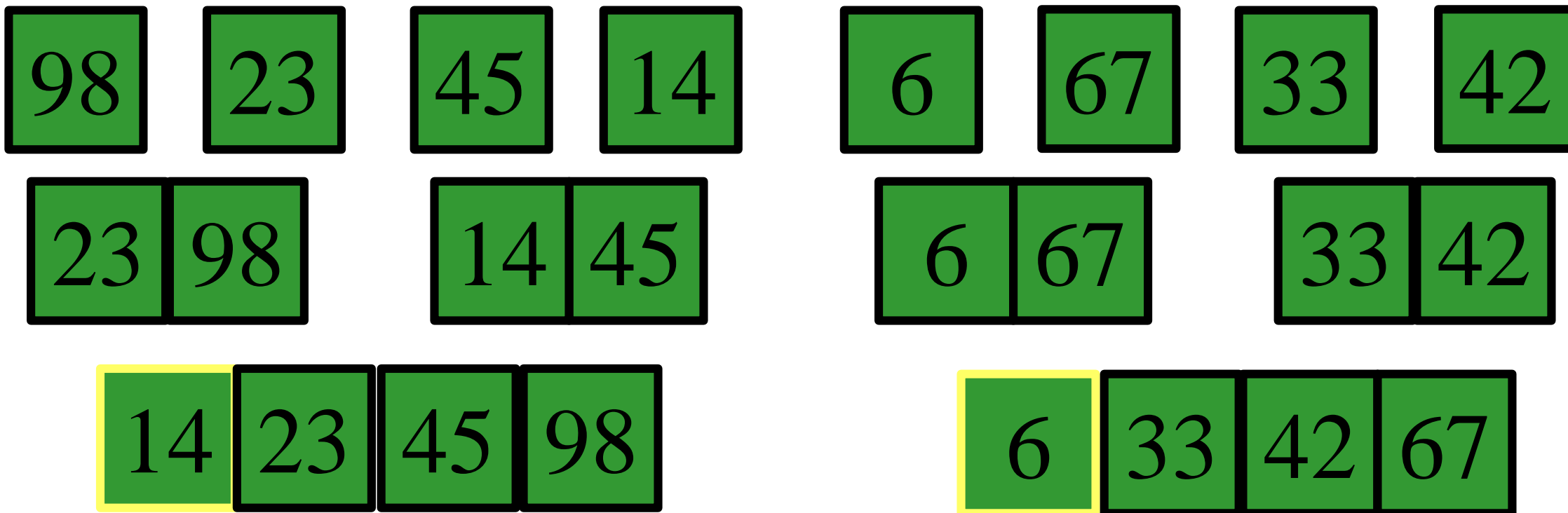
문제 8



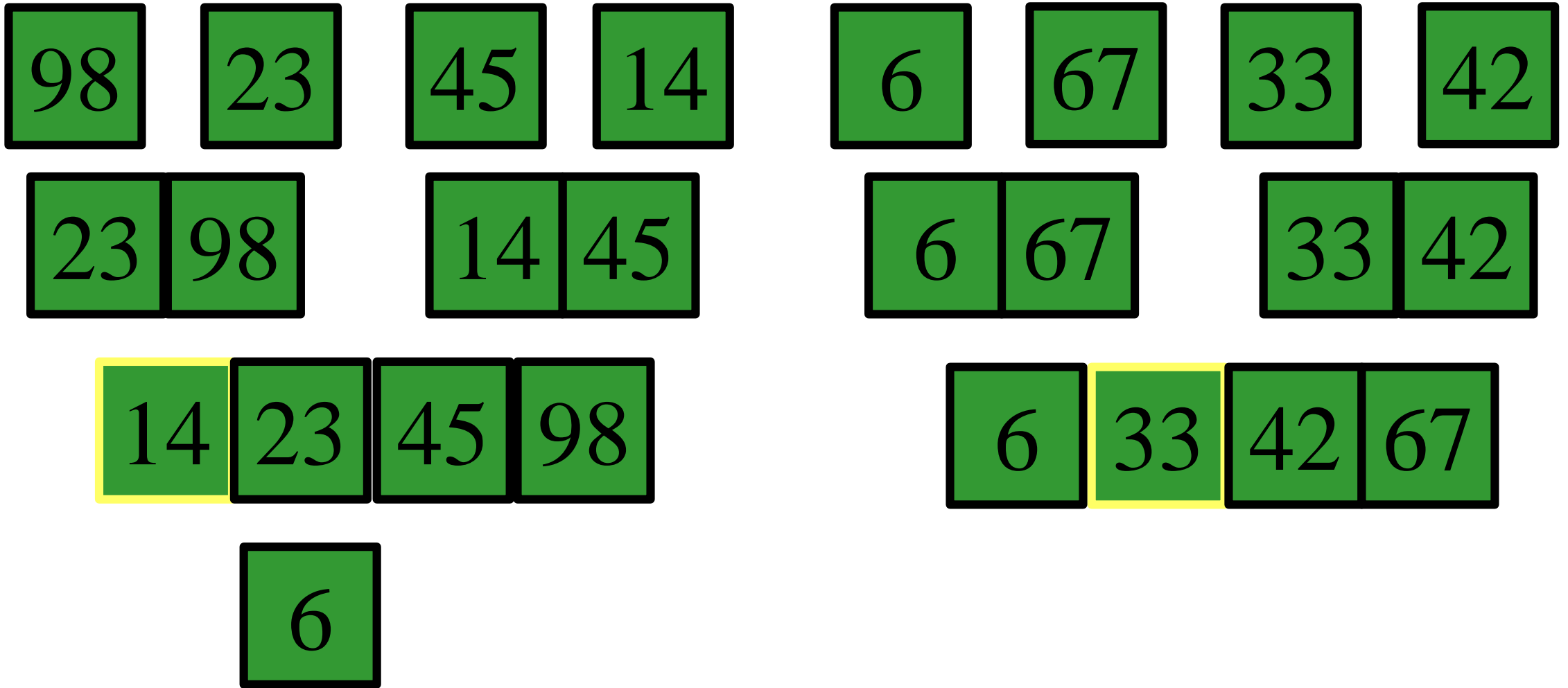
문제 8



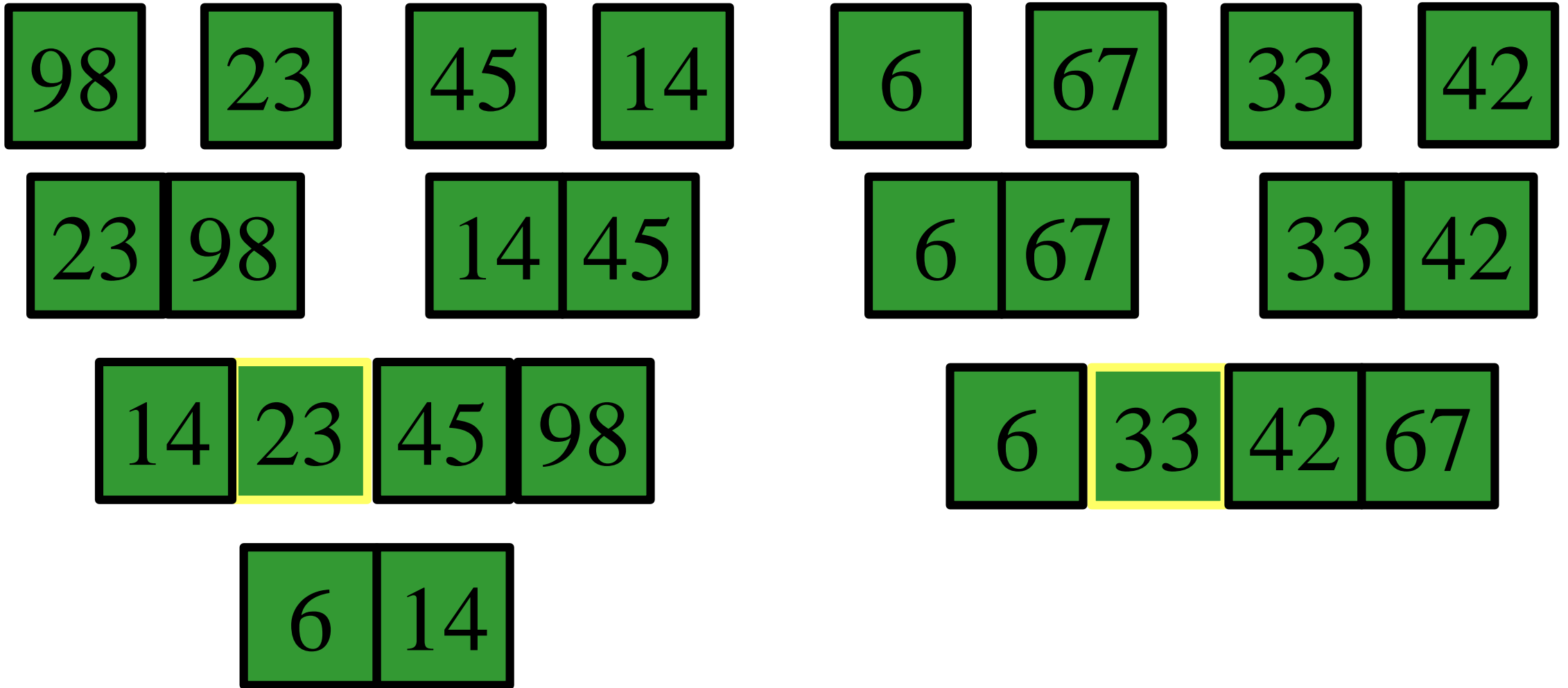
문제 8



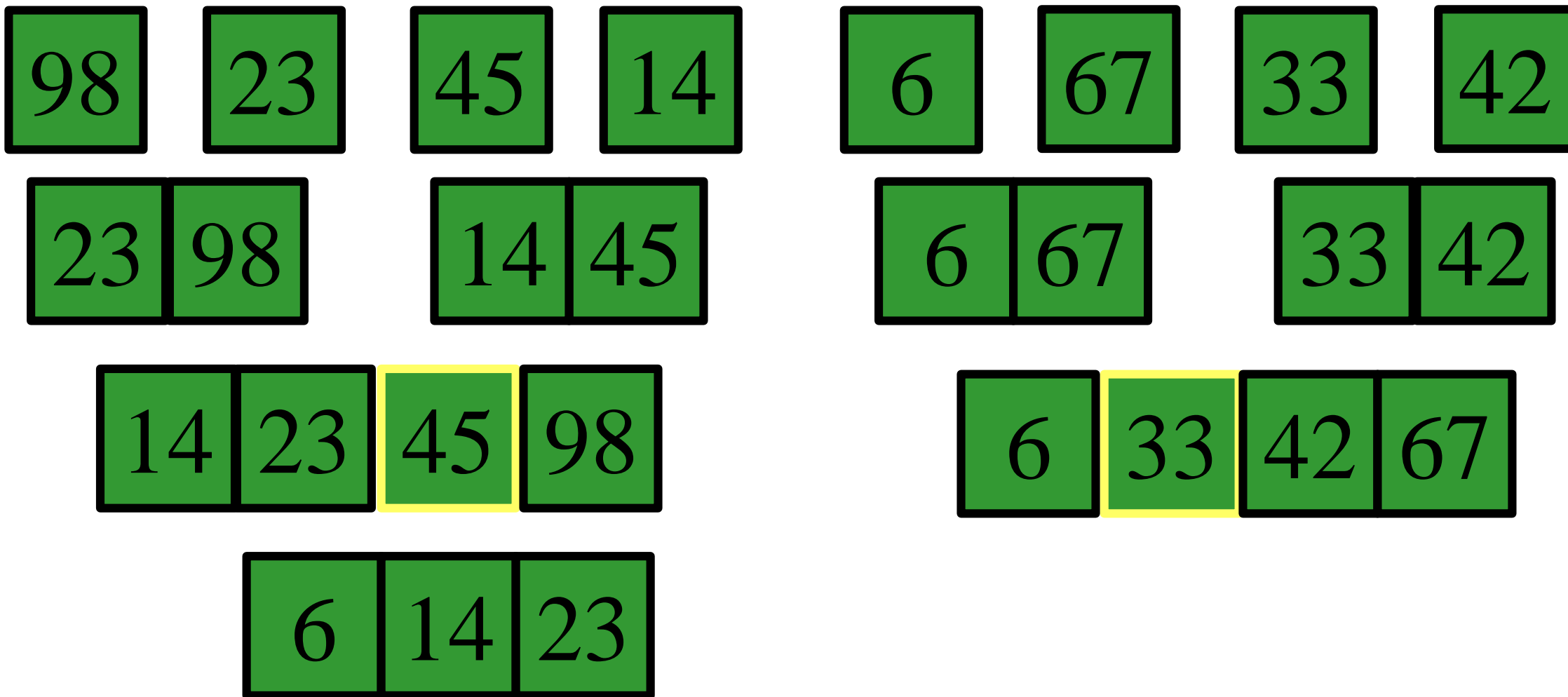
문제 8



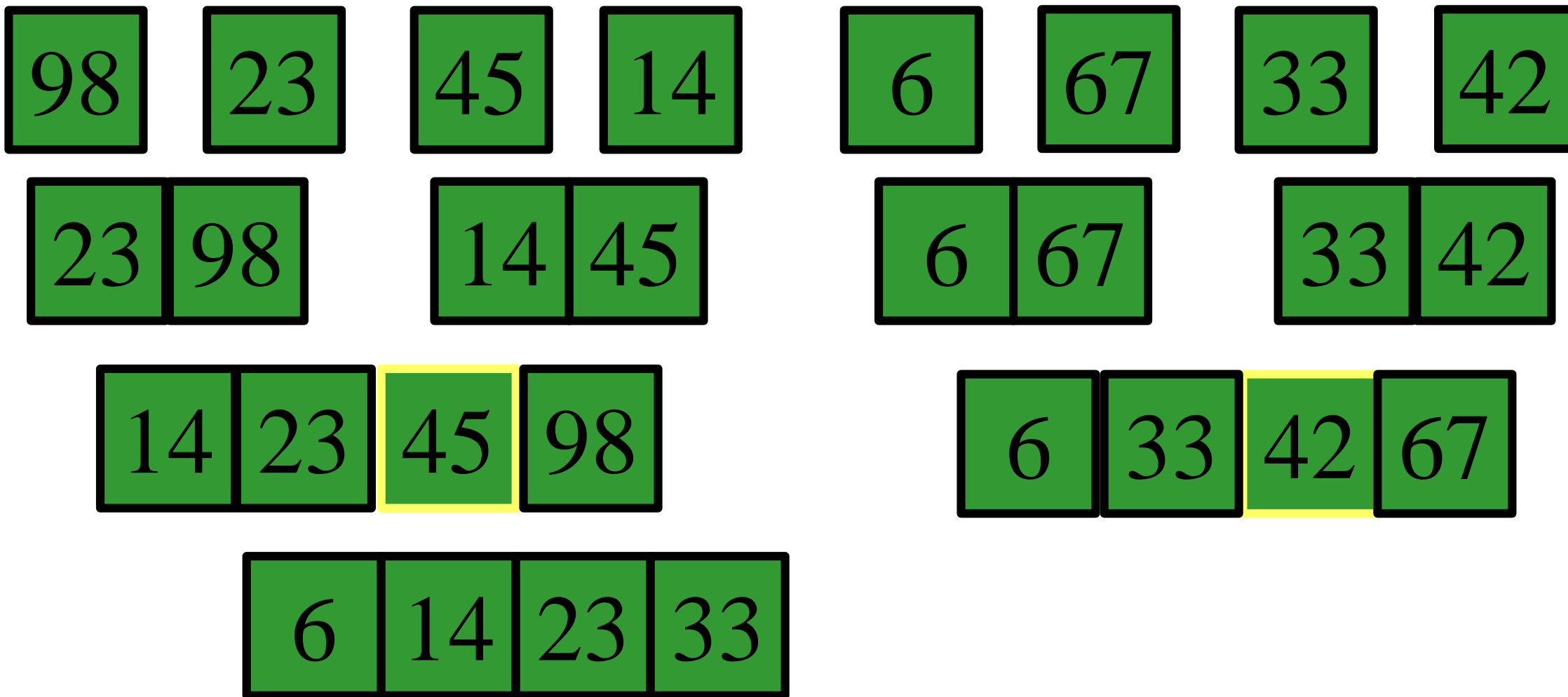
문제 8



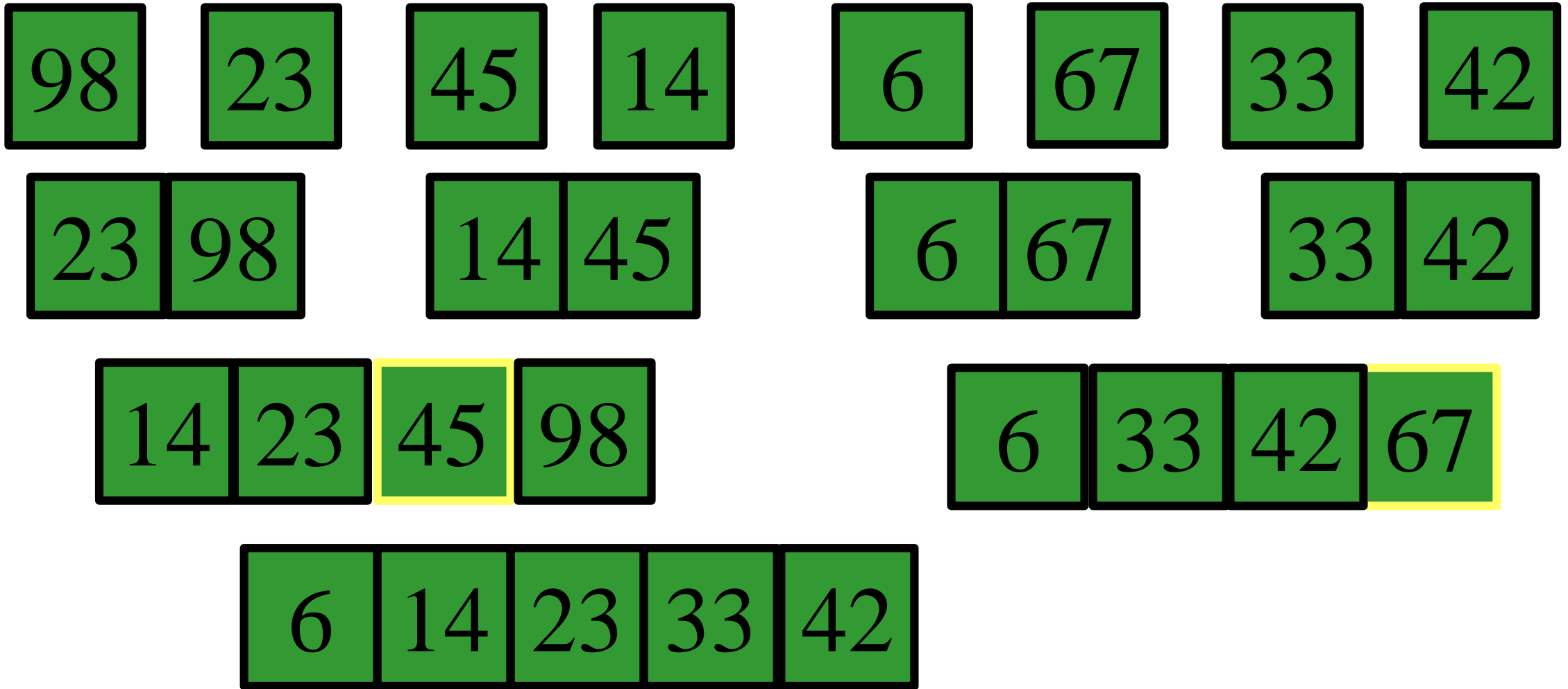
문제 8



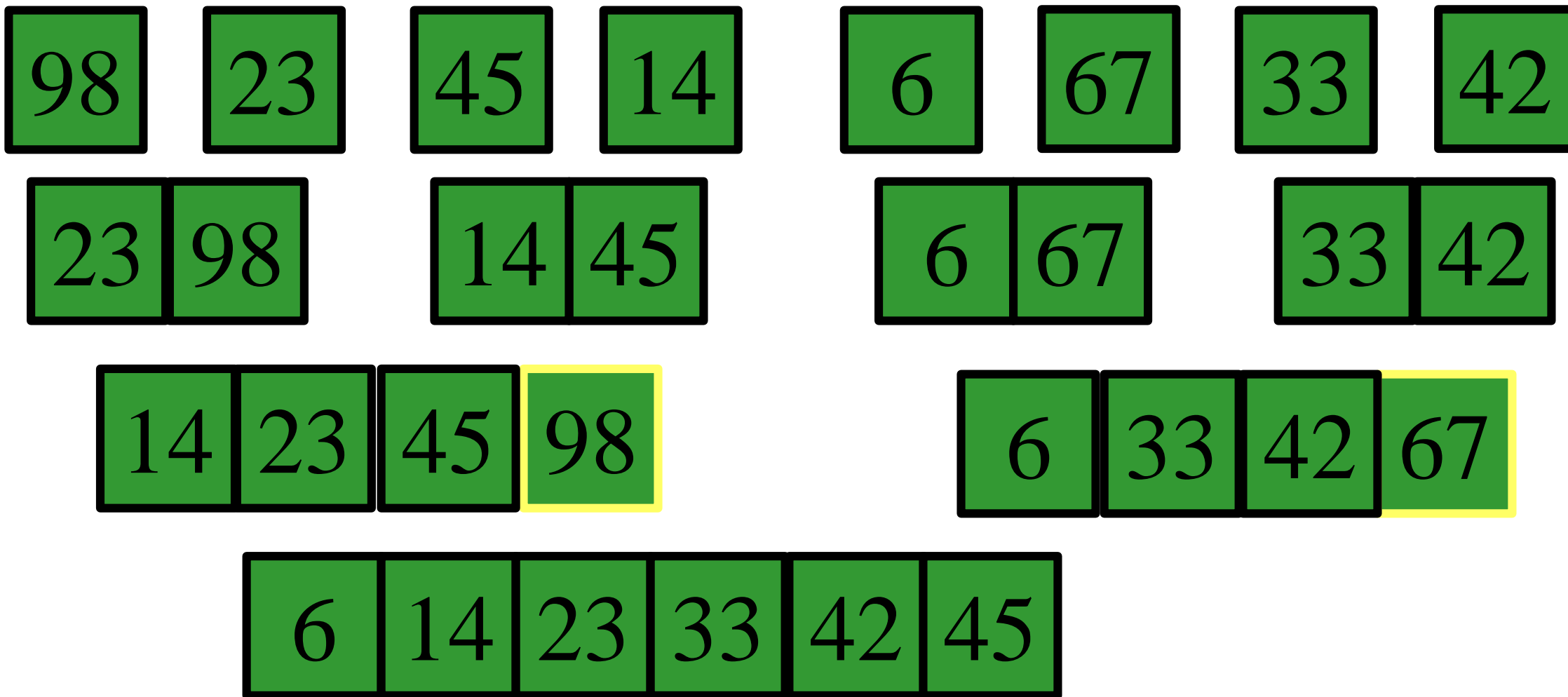
문제 8



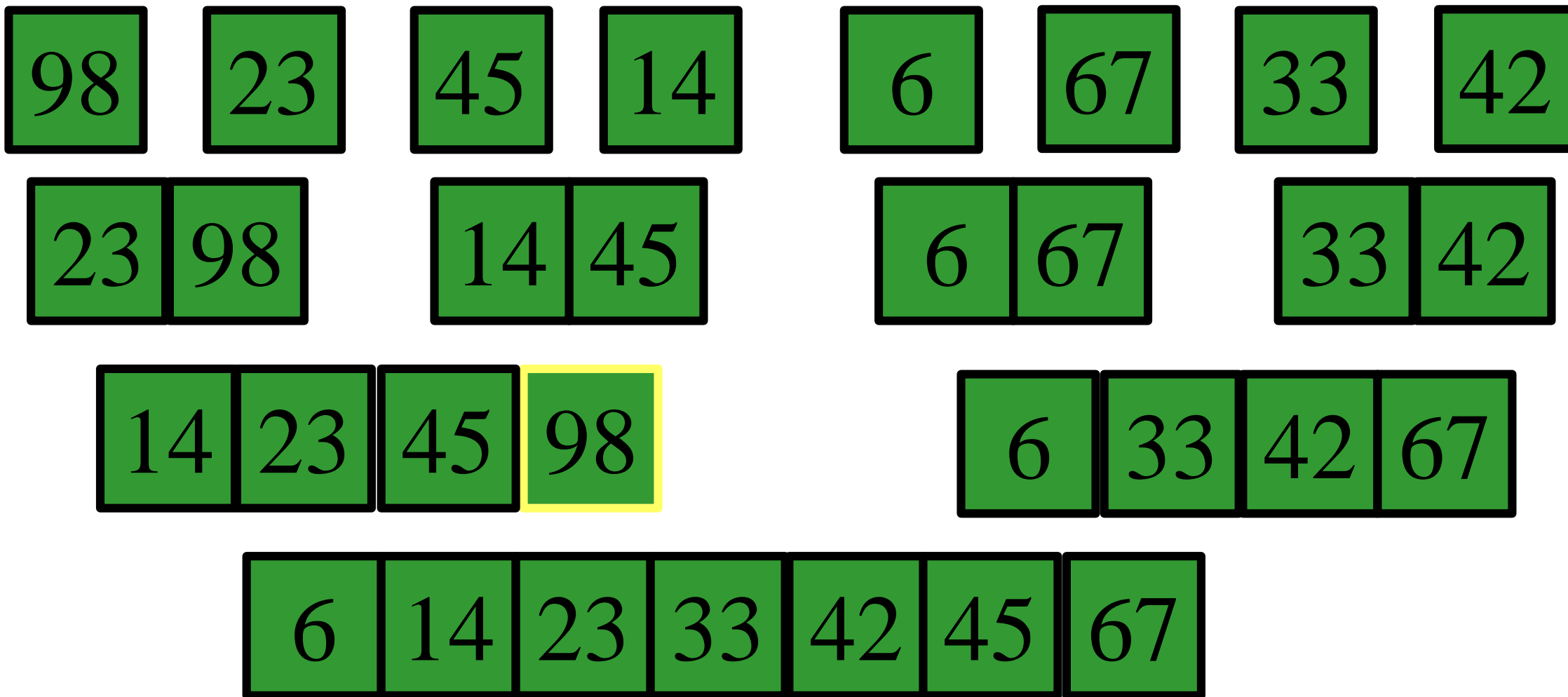
문제 8



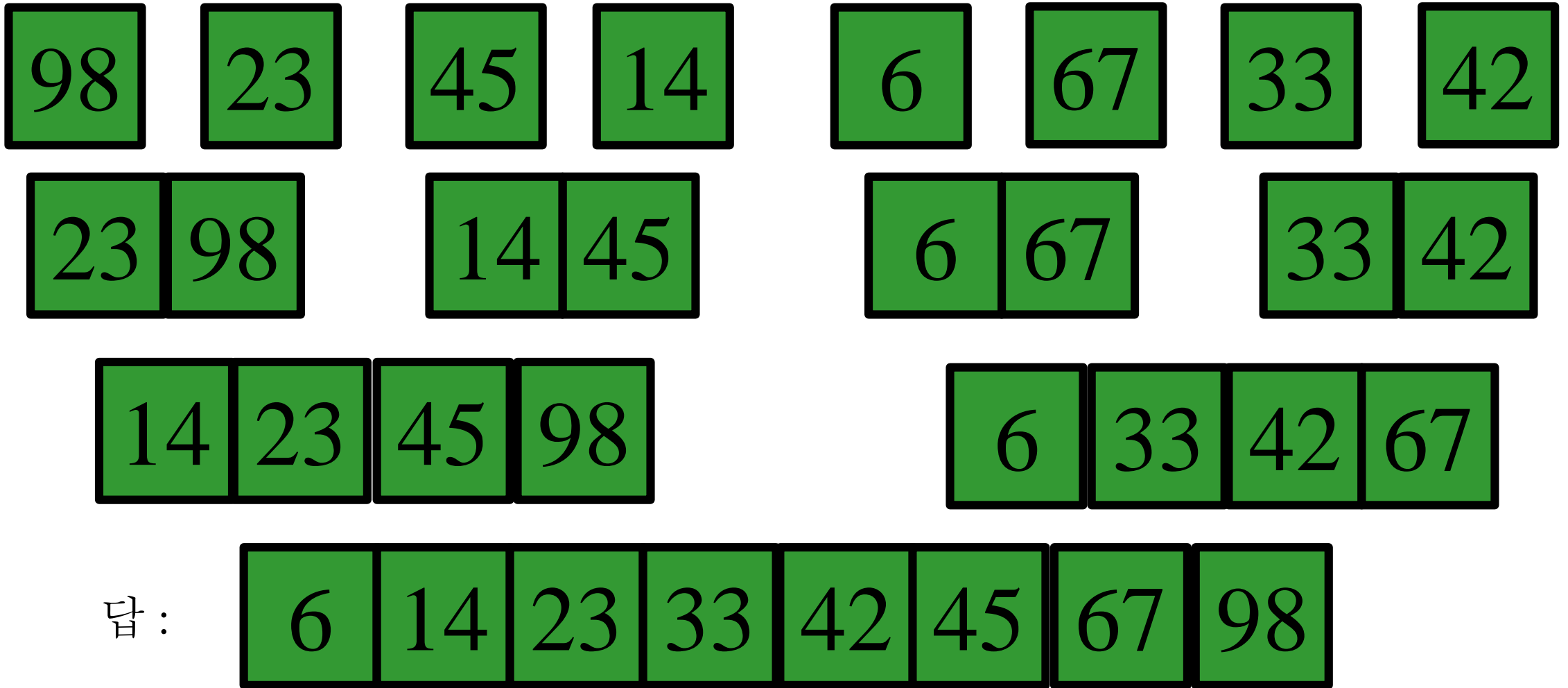
문제 8



문제 8

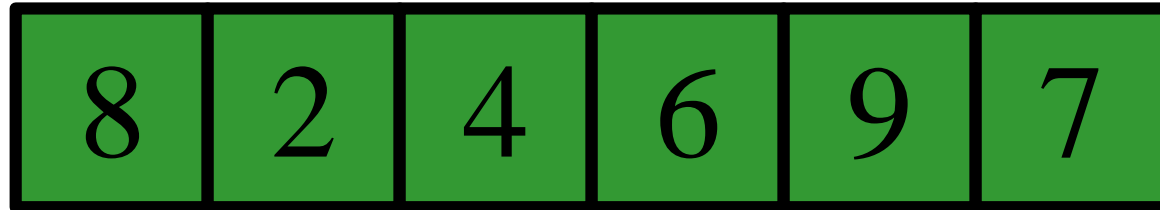


문제 8



문제 5

문제 : 합병정렬을 사용하여 정렬하는 과정을 보여라. (Sort the keys using merge sort. Show sorting process.)
keys={8, 2, 4, 6, 9, 7}



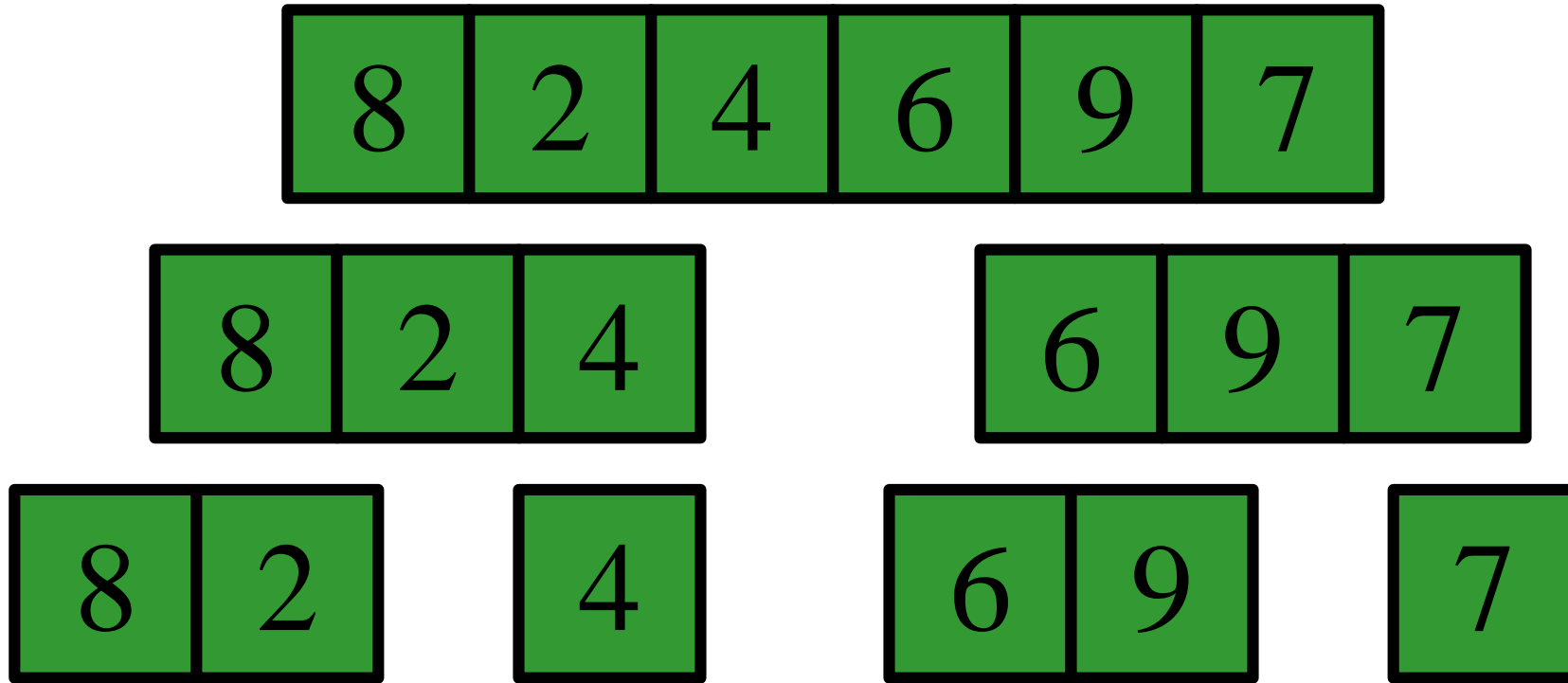
문제 5

8	2	4	6	9	7
---	---	---	---	---	---

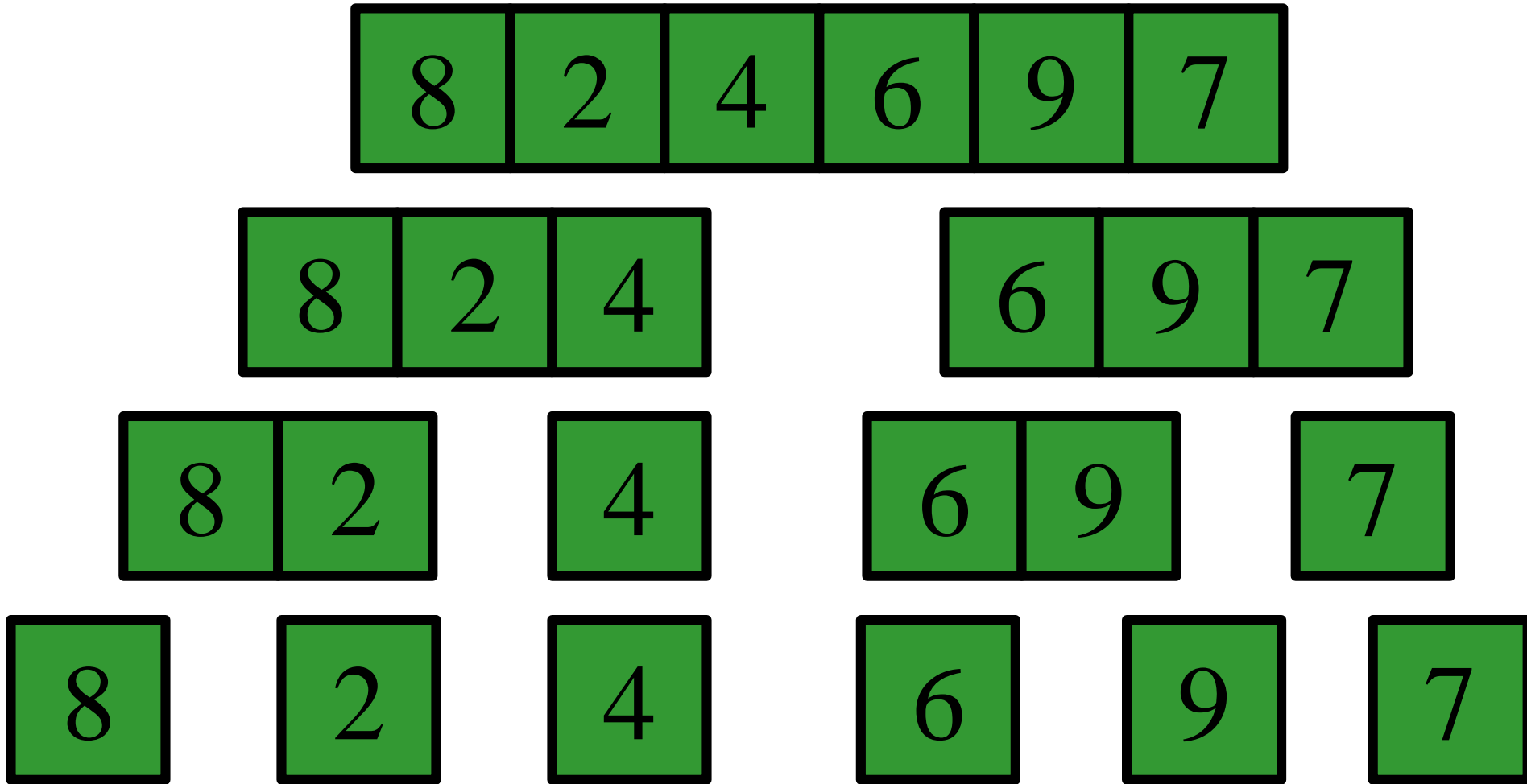
8	2	4
---	---	---

6	9	7
---	---	---

문제 5



문제 5



문제 5

8

2

4

6

9

7

문제 5

8

2

4

6

9

7

2

문제 5

8

2

4

6

9

7

2	8
---	---

문제 5

8

2

4

6

9

7

2 8

문제 5

8

2

4

6

9

7

2

8

2

문제 5

8

2

4

6

9

7

2 8

2 4

문제 5

8

2

4

6

9

7

2	8
---	---

2	4	8
---	---	---

문제 5

8

2

4

6

9

7

2 8

6

2 4 8

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

6

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

6 7

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

6 7 9

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

6 7 9

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

6 7 9

2

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

6 7 9

2 4

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

6 7 9

2 4 6

문제 5

8

2

4

6

9

7

2 8

6 9

2 4 8

6 7 9

2 4 6 7

문제 5

8

2

4

6

9

7

2 8

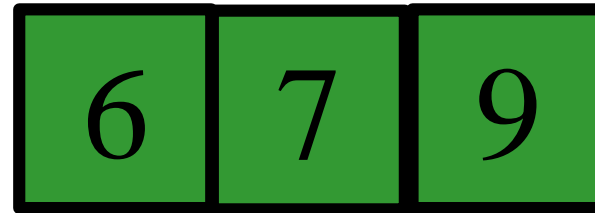
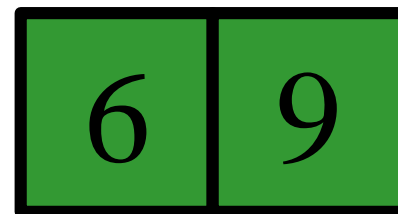
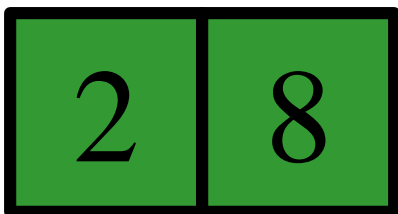
6 9

2 4 8

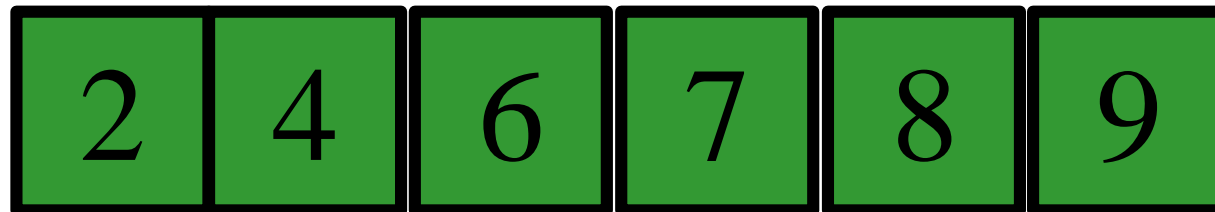
6 7 9

2 4 6 7 8

문제 5



답 :



문제 6

문제 : 쉘정렬을 사용하여 정렬하는 과정을 보여라. ($h=3$, $h=1$ 을 사용하라) (Sort the keys using shell sort. Show sorting process. Use $h=3$, $h=1$.) keys={8, 2, 4, 6, 9, 7}

문제 6

H=3

8	2	4	6	9	7
6	2	4	8	9	7
6	2	4	8	9	7
6	2	4	8	9	7

문제 6

H=1

6	2	4	8	9	7
2	6	4	8	9	7
2	4	6	8	9	7
2	4	6	8	9	7
2	4	6	8	9	7
2	4	6	8	9	7
2	4	6	8	7	9
2	4	6	8	7	9
2	4	6	8	7	9
2	4	6	8	7	9
2	4	6	7	8	9
2	4	6	7	8	9

문제 7

문제 : 내부정렬 알고리즘과 외부정렬 알고리즘의 차이를 적어라.
(Write the difference between internal sorting and external sorting.)







답 : 내부 정렬 알고리즘은 정렬할 자료를 메인 메모리에 모두 올려서 정렬하는 방식이고 외부정렬은 정렬할 자료를 보조 기억장치로 할당해 보조기억장치에서 정렬하는 방식

내부정렬이 정렬속도가 더 빠르지만 정렬해야하는 자료의 용량이 클수록 외부정렬 알고리즘이 유리

문제 8

문제 : 원소 23, 123, 234, 507, 3, 43, 253, 287을 기수 정렬을 사용하여 정렬하는 과정을 보 여라. (Sort the keys using radix sort. Show sorting process.)

문제 8

			
023	023	003	003
123	123	023	023
234	003	123	043
527	 043	 527	 123
003	253	234	234
043	234	043	253
253	527	253	287
287	287	287	527

2017년도 9번

Q. 329, 457, 657, 839, 436, 720, 355를 기수정렬로 해결하라. 그 과정을 보여라.

A.

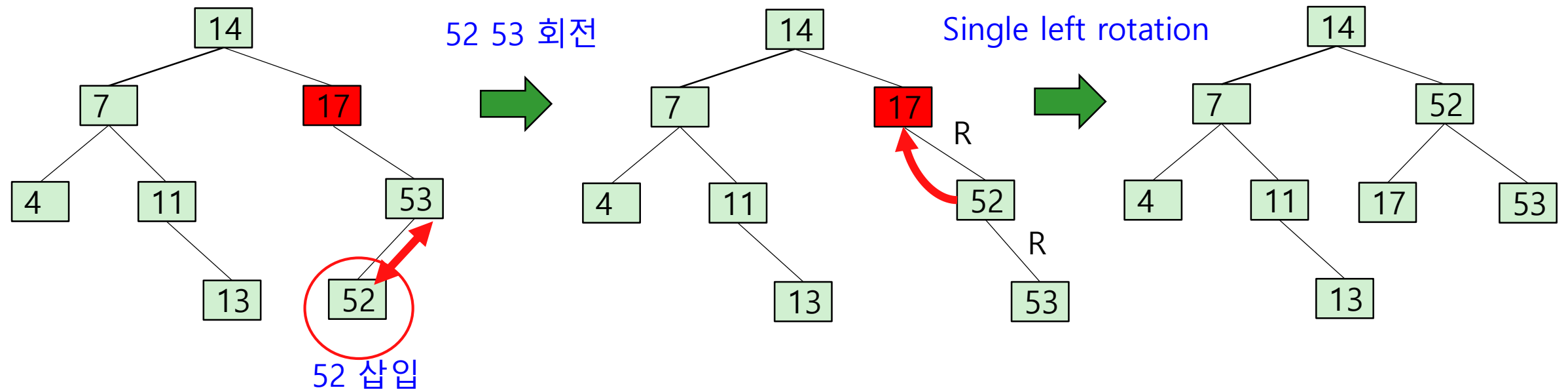
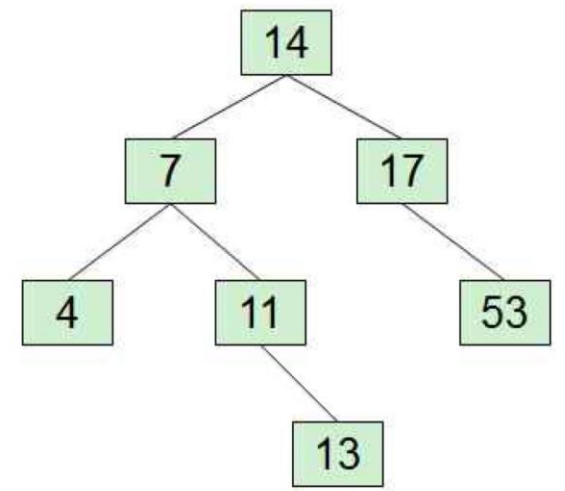
		↓		↓		↓
329		720		720		329
457		355		329		355
657		436		436		436
839	→	457	→	839	→	457
436		657		355		657
720		329		457		720
355		839		657		839



2017년도 10번

Q. 그림 1의 AVL 트리에 52를 삽입하였다. 결과되는 트리를 그려라.

A.



2017년도 11번

Q. 스펙트럼 트리에 대하여 설명하라.

A. 이진 탐색 트리 중 하나로 완전 균형이 아니고 한번 탐색되었던 원소는 다시 탐색 되기 쉽다는 가정을 기반으로 두어 삽입, 삭제, 탐색 원소를 root로 가져와 다음 탐색이 빠르도록 한다.

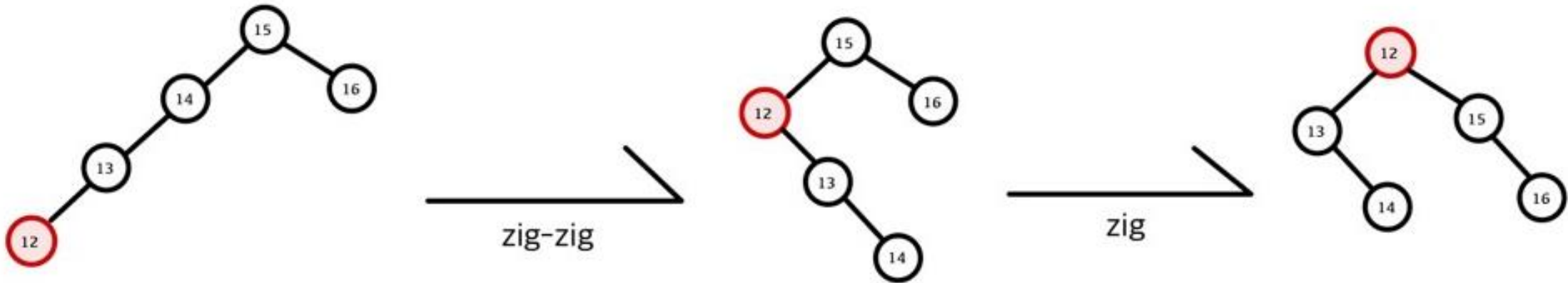
연산(탐색, 삽입, 삭제 등) 하는 시간은 $O(\log n)$ 에 비례하며 최악의 경우 $O(n)$ 의 시간에 비례한다.



2017년도 12번

Q. 그림 2의 스플레이 트리에서 12를 탐색한 이후의 스플레이 트리를 그려라.

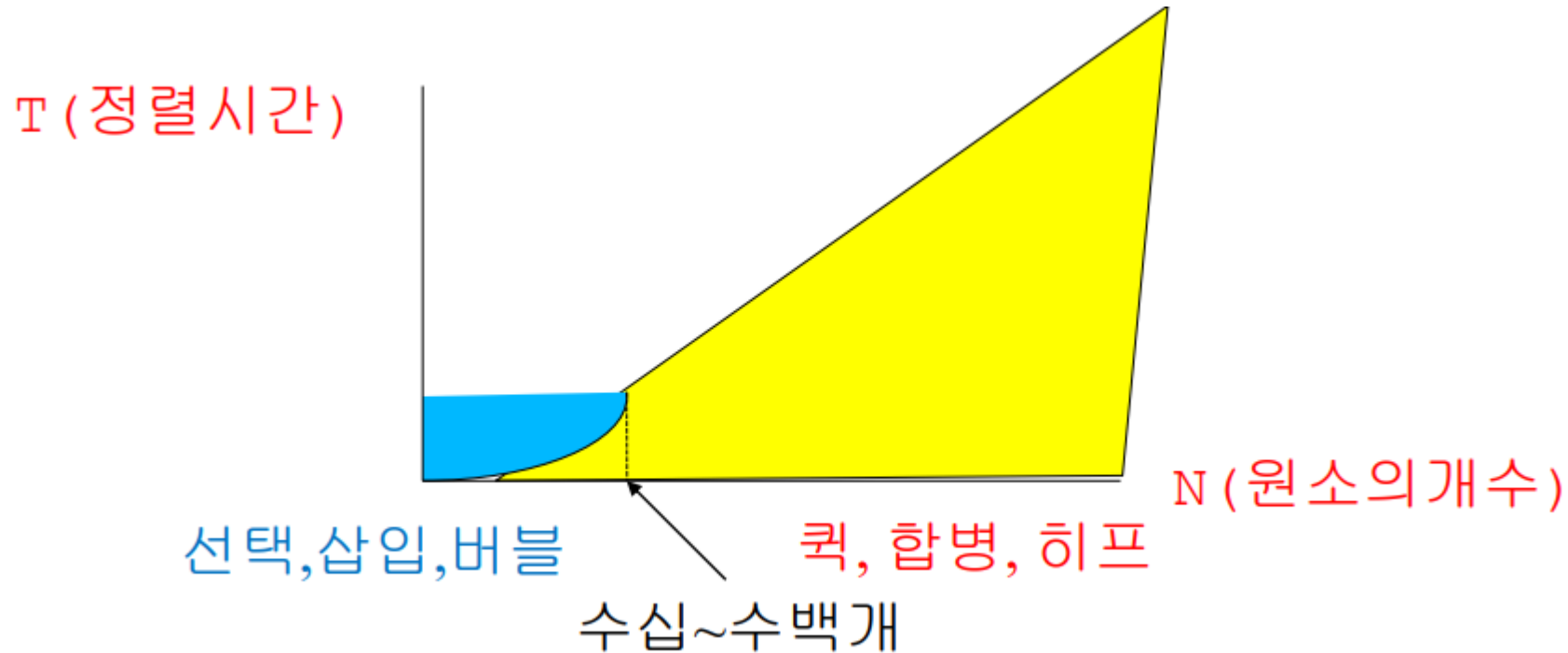
A. 12 위에 아빠, 할아버지 원소가 있어 zig-zig 해서 아래 가운데 그림처럼 트리가 만들고 12 위에 아직 아빠원소가 있기 때문에 한번 더 zig 해주어 아래 마지막 그림과 같이 트리의 root에 12를 올려주면 탐색이 끝난다.



(2020)문제9번

◆ 답 선택, 삽입, 버블

수십 개의 데이터를 정렬하고 싶다. 어떤 정렬 알고리즘을 사용할 것인지 결정하고, 그 이유를 설명하라.



(2020)문제10번

◆ 답: 제자리정렬 알고리즘이다.

김진 교수가 새로운 정렬 알고리즘(FastSort)을 개발하였다. 이 정렬 알고리즘은 제자리정렬 알고리즘인가? 그 이유를 적어라.

```
FastSort(a[], n){  
    int b[n]=new int[n];  
    for(i←n-1; i≥0; i←i-1)  
        for(j←0; j<i; j←j + 1){  
            .....  
            b[i]=a[tmp];  
            .....  
        }  
    }  
}
```

정렬하고자하는 원소의 개수에 비례하는 임시공간이 필요하므로 제자리정렬 알고리즘이 아니다.



2020년도 12번

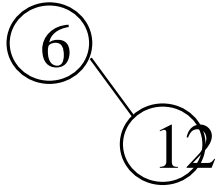
Q. 공백 AVL트리에 원소 6, 12, 9, 7, 2, 11의 원소를 순서대로 삽입할 때 생성되는 최종 AVL 트리를 그려라.

A.

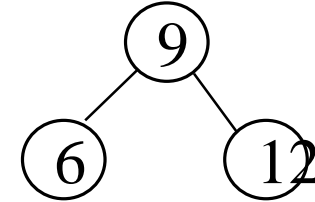
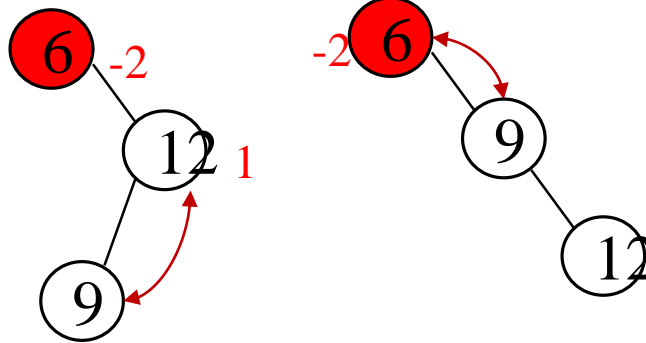
Insert 6



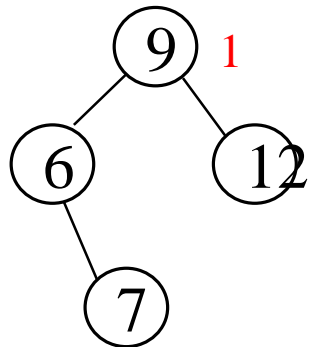
Insert 12



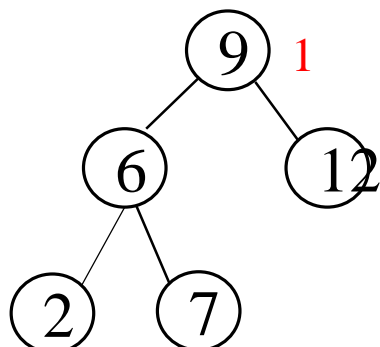
Insert 9



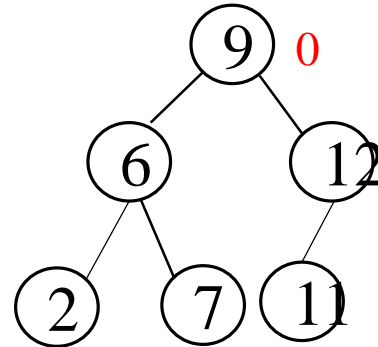
Insert 7



Insert 2



Insert 11

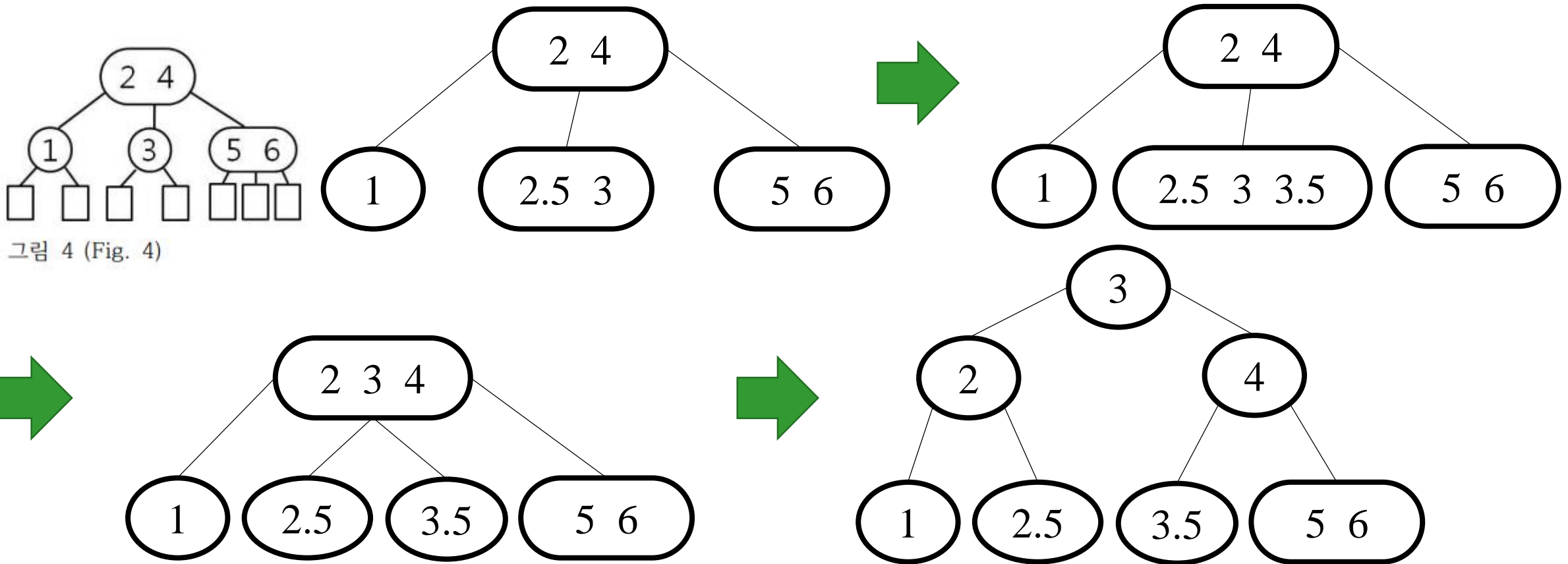


문제 13

- ◆ 2-3 트리가 스플레이 트리와 다른 점을 설명하라. (Explain the difference between 2-3 tree and splay tree)
- ◆ 답: 2-3트리는 스플레이 트리와 달리 자식을 3개까지 가질 수 있으며 단말 노드들은 같은 레벨을 가진다.

문제 14-1

- 다음 2-3 트리에서 2.5와 3.5를 삽입하였을 때 결과로 나타나는 2-3 트리를 그려라. (Draw 2-3 tree after 2.5 and 3.5 inserted in Fig. 4)



문제 15

- ◆ 2-3-4 트리가 2-3 트리와 비교하여 가지는 장점을 설명하라.
(Compare 2-3-4 tree and 2-3 tree. Explain the advantages of 2-3-4 tree.)
- ◆ 답: 2-3-4트리는 2-3트리와 달리 후진 분할이 일어나지 않아 삽입과 삭제가 한번의 패스로 끝나기 때문에 더 효율적이다.

문제 16

- 다음 2-3-4 트리(그림 3)에서 1을 삽입하였을 때의 결과로 나타나는 2-3-4 트리를 그려라. (Draw 2-3-4 tree after element 1 is inserted in Fig. 3)

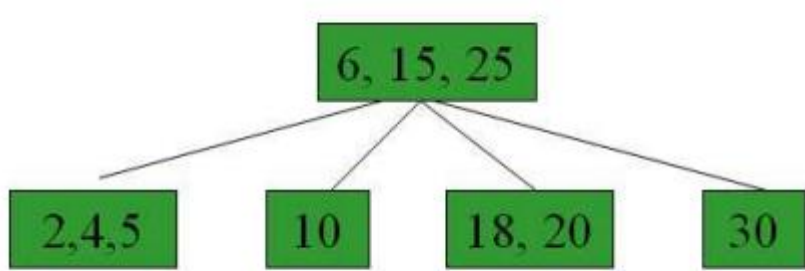
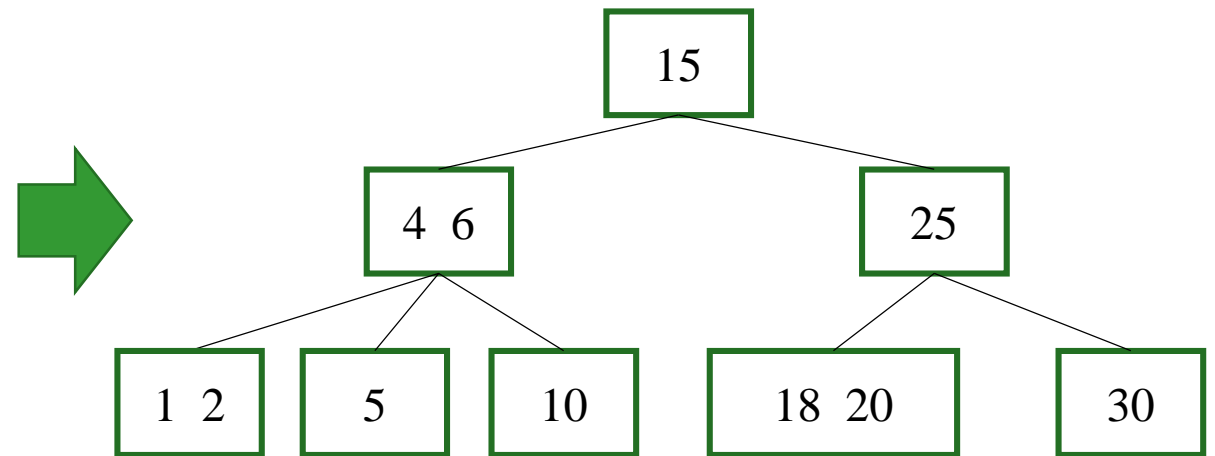
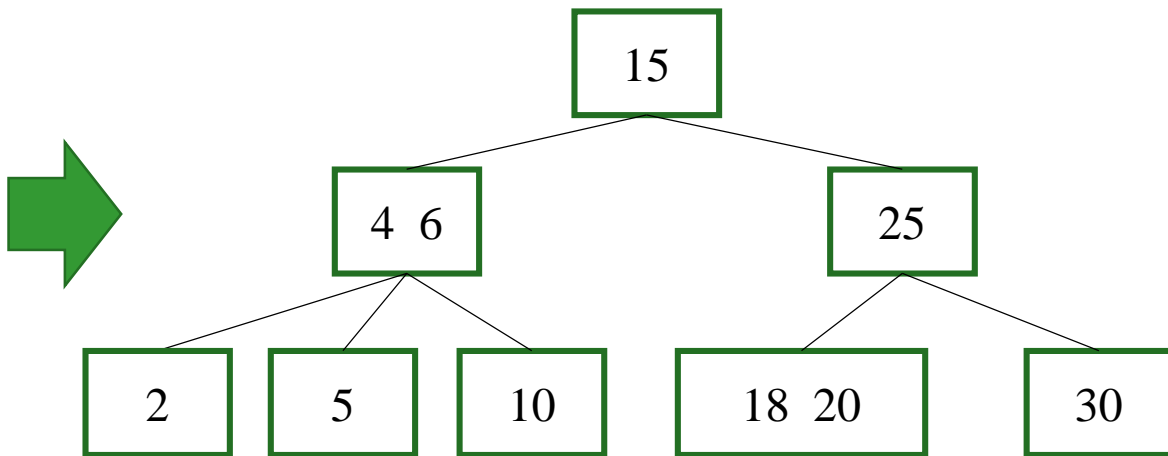
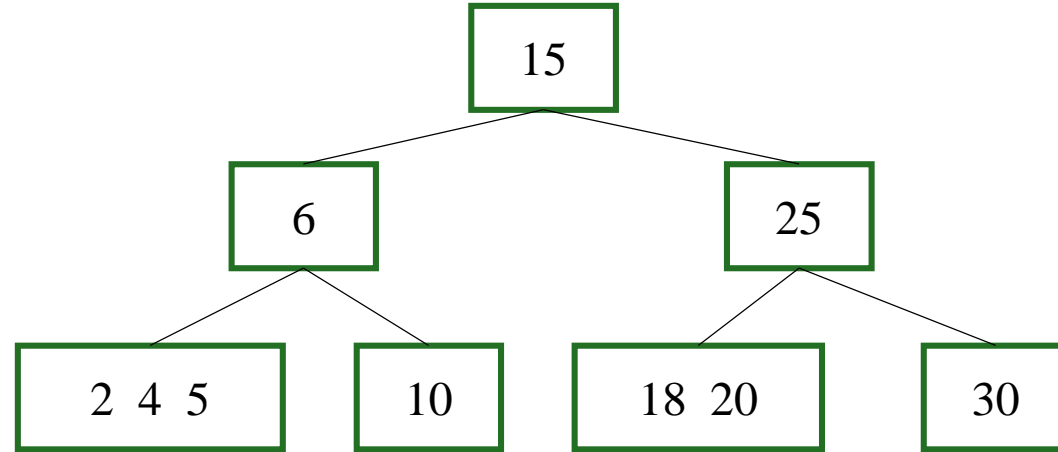


그림 3 (Fig. 3)

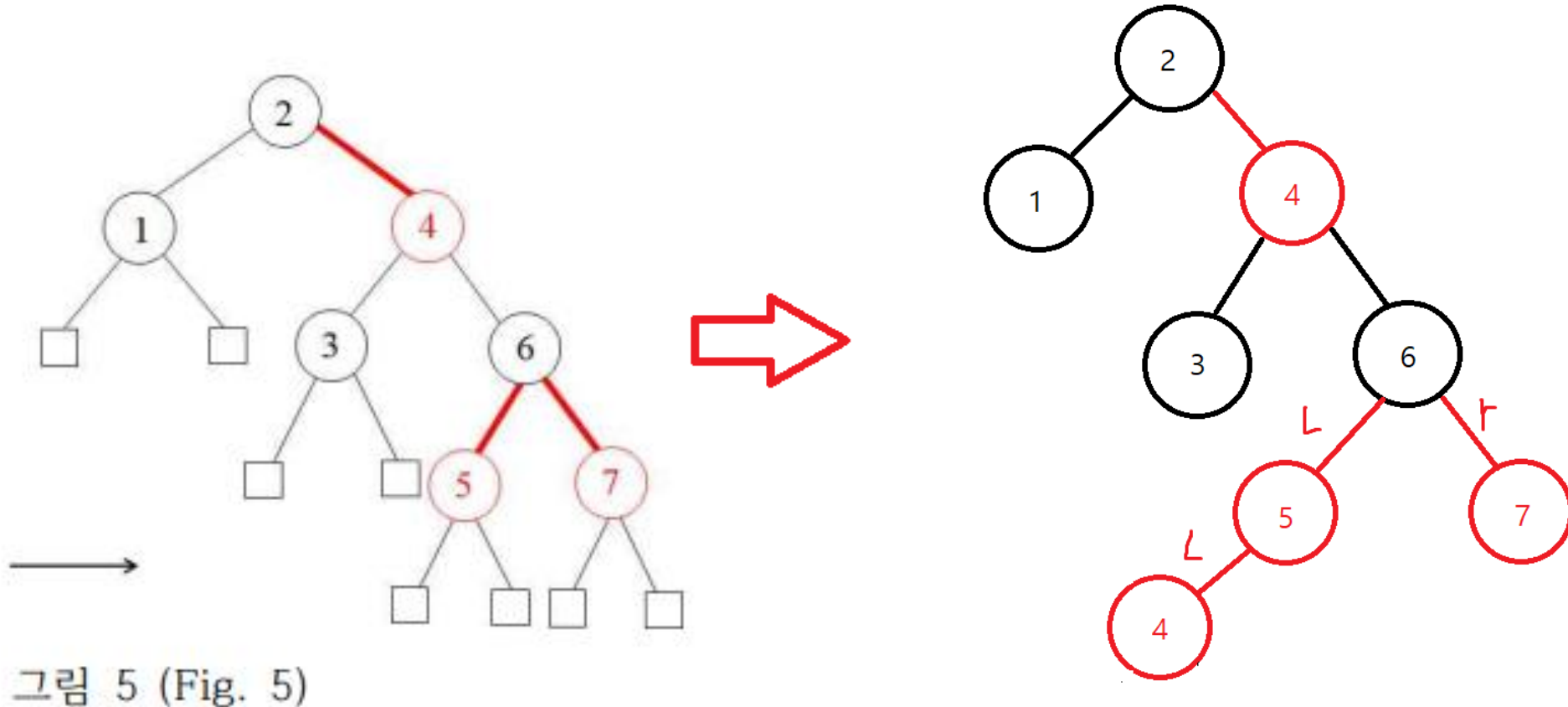


문제 17

- ◆ 레드 블랙 트리가 2-3-4 트리와 비교하여 가지는 장점을 설명하라.
(Compare red black tree and 2-3-4 tree. Explain the advantages of red-black tree)
- ◆ 답: 레드 블랙 트리는 2-3-4 트리를 이진 탐색트리로 표현한 것이기 때문에 노드의 구조가 더 단순하고 탐색시간도 빠르다. 또한 언제 회전에 의해 균형을 잡아야 하는지가 쉽게 판별된다.

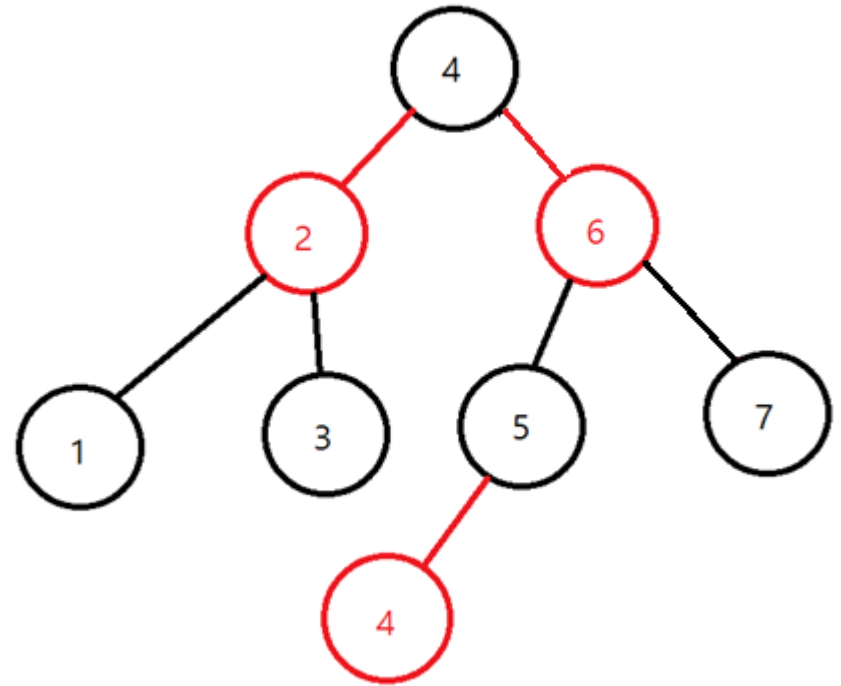
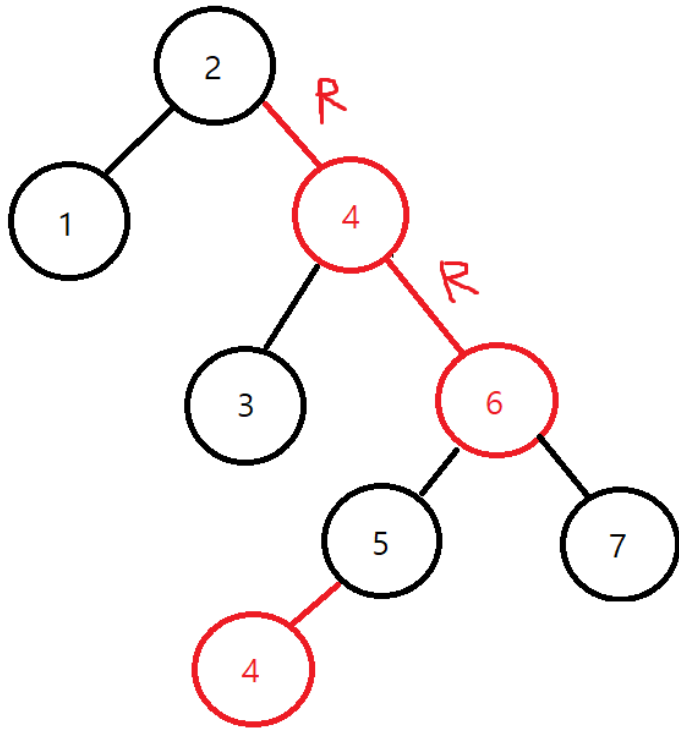
문제 18-1

- ◆ 다음 레드 블랙 트리에 4를 삽입하였을 때 결과 레드 블랙 트리를 그려라. (Draw red black tree after 4 inserted in Fig. 5).



문제 18-2

- ◆ 다음 레드 블랙 트리에 4를 삽입하였을 때 결과 레드 블랙 트리를 그려라. (Draw red black tree after 4 inserted in Fig. 5).



알고리즘 기말고사(2020-2)

(D조)

<http://smartlead.hallym.ac.kr>

Instructor: Jin Kim
010-6267-8189(033-248-2318)
jinkim@hallym.ac.kr

Office Hours:

20205270 한주영

20195224 이주영

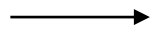
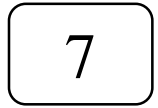
20185285 임수빈

20185266 이규석

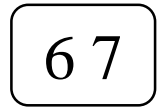
문제 13

- ◆ 2-3 트리에 7, 6, 5, 4, 3, 2, 1을 차례로 삽입한 최종 트리를 그려라.
(Draw 2-3 tree after 7, 6, 5, 4, 3, 2, 1 inserted one by one)

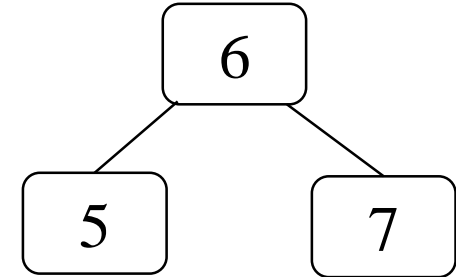
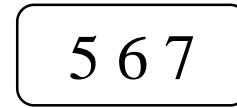
7 삽입



6 삽입

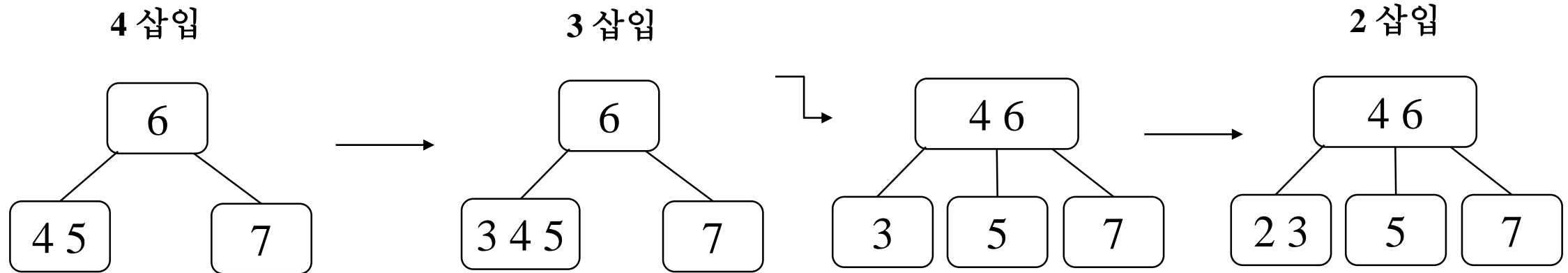


5 삽입



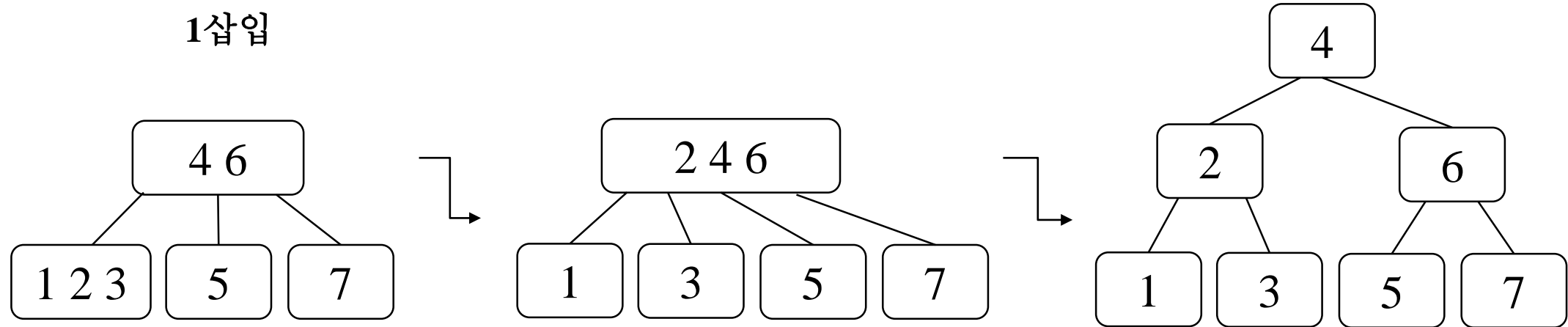
문제 13

- ◆ 2-3 트리에 7, 6, 5, 4, 3, 2, 1을 차례로 삽입한 최종 트리를 그려라.
(Draw 2-3 tree after 7, 6, 5, 4, 3, 2, 1 inserted one by one)



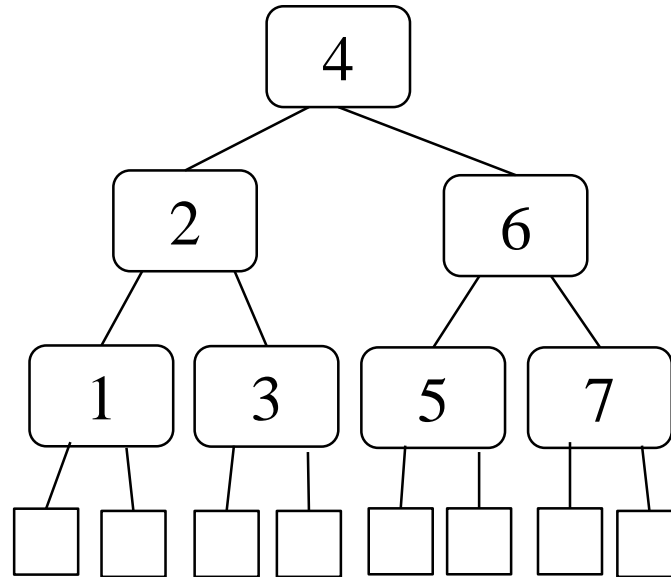
문제 13

- ◆ 2-3 트리에 7, 6, 5, 4, 3, 2, 1을 차례로 삽입한 최종 트리를 그려라.
(Draw 2-3 tree after 7, 6, 5, 4, 3, 2, 1 inserted one by one)



문제 13

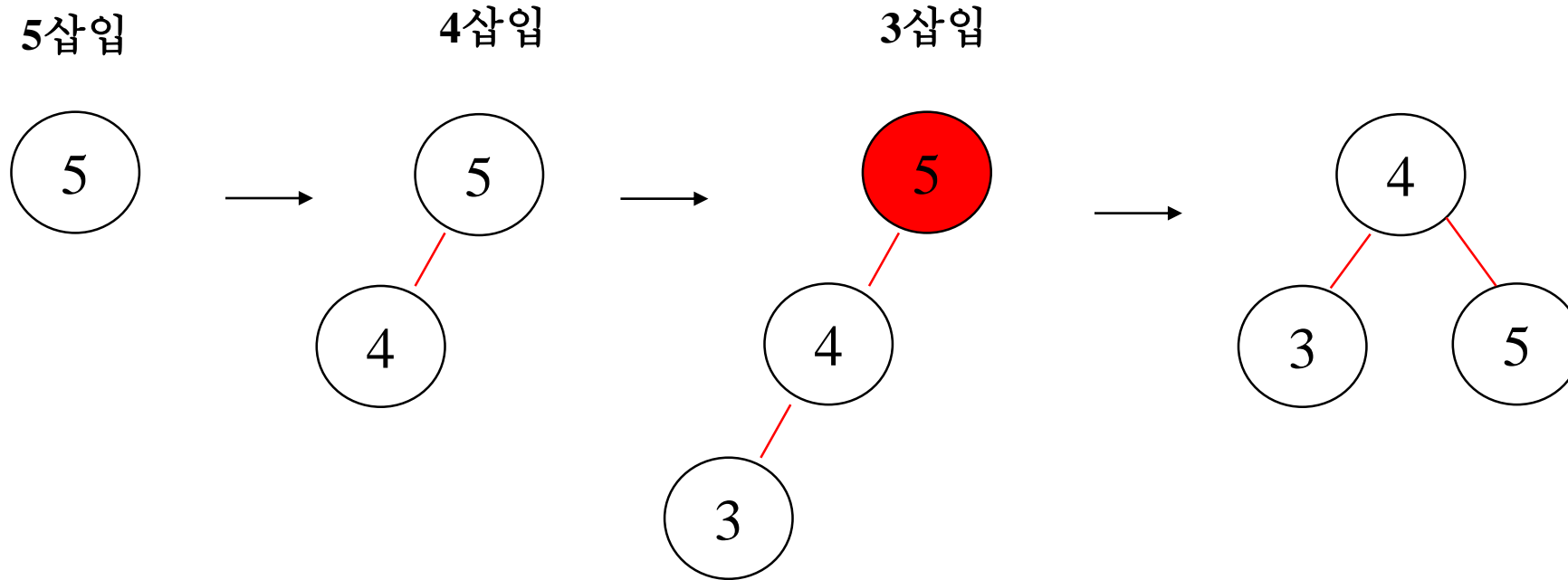
- ◆ 2-3 트리에 7, 6, 5, 4, 3, 2, 1을 차례로 삽입한 최종 트리를 그려라.
(Draw 2-3 tree after 7, 6, 5, 4, 3, 2, 1 inserted one by one)



<최종결과>

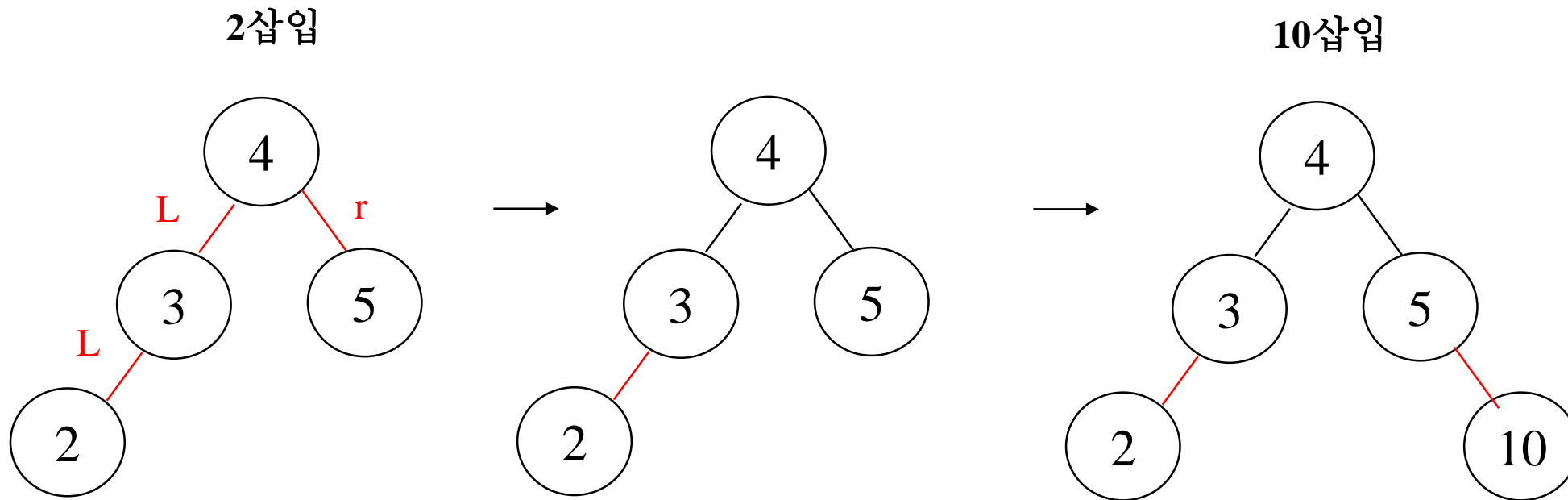
문제 14

- ◆ 공백 레드 블랙 트리에서 키 5, 4, 3, 2, 10, 9, 8을 기술한 순서대로 삽입한 결과를 그려라. (Draw red-black tree after 5, 4, 3, 2, 10, 9, 8 inserted one by one.)



문제 14

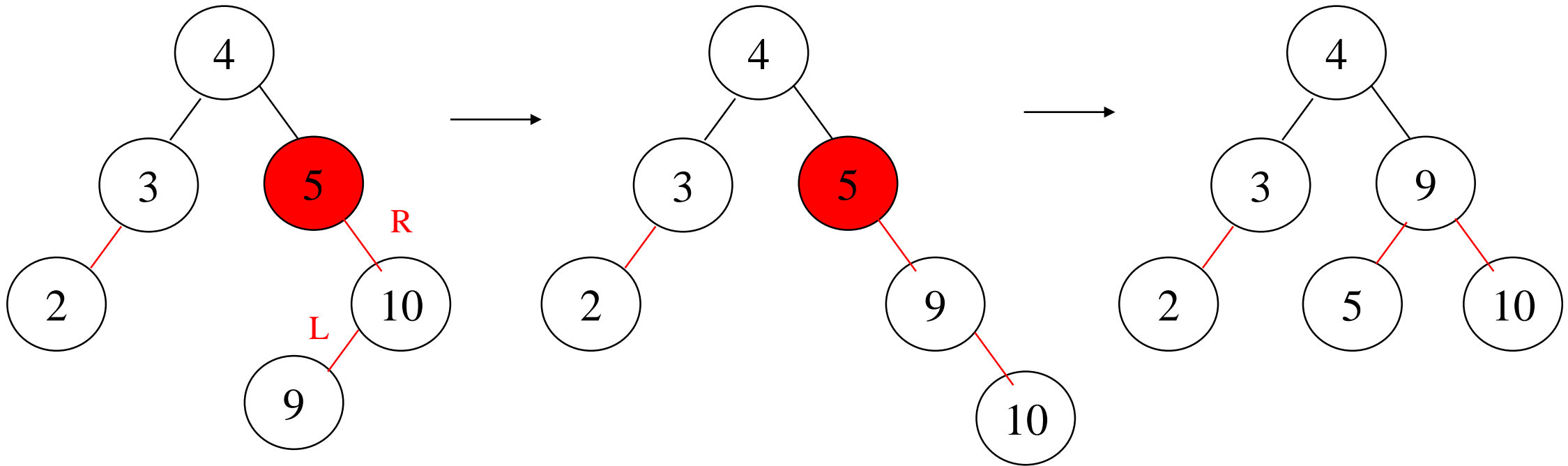
- ◆ 공백 레드 블랙 트리에서 키 5, 4, 3, 2, 10, 9, 8을 기술한 순서대로 삽입한 결과를 그려라. (Draw red-black tree after 5, 4, 3, 2, 10, 9, 8 inserted one by one.)



문제 14

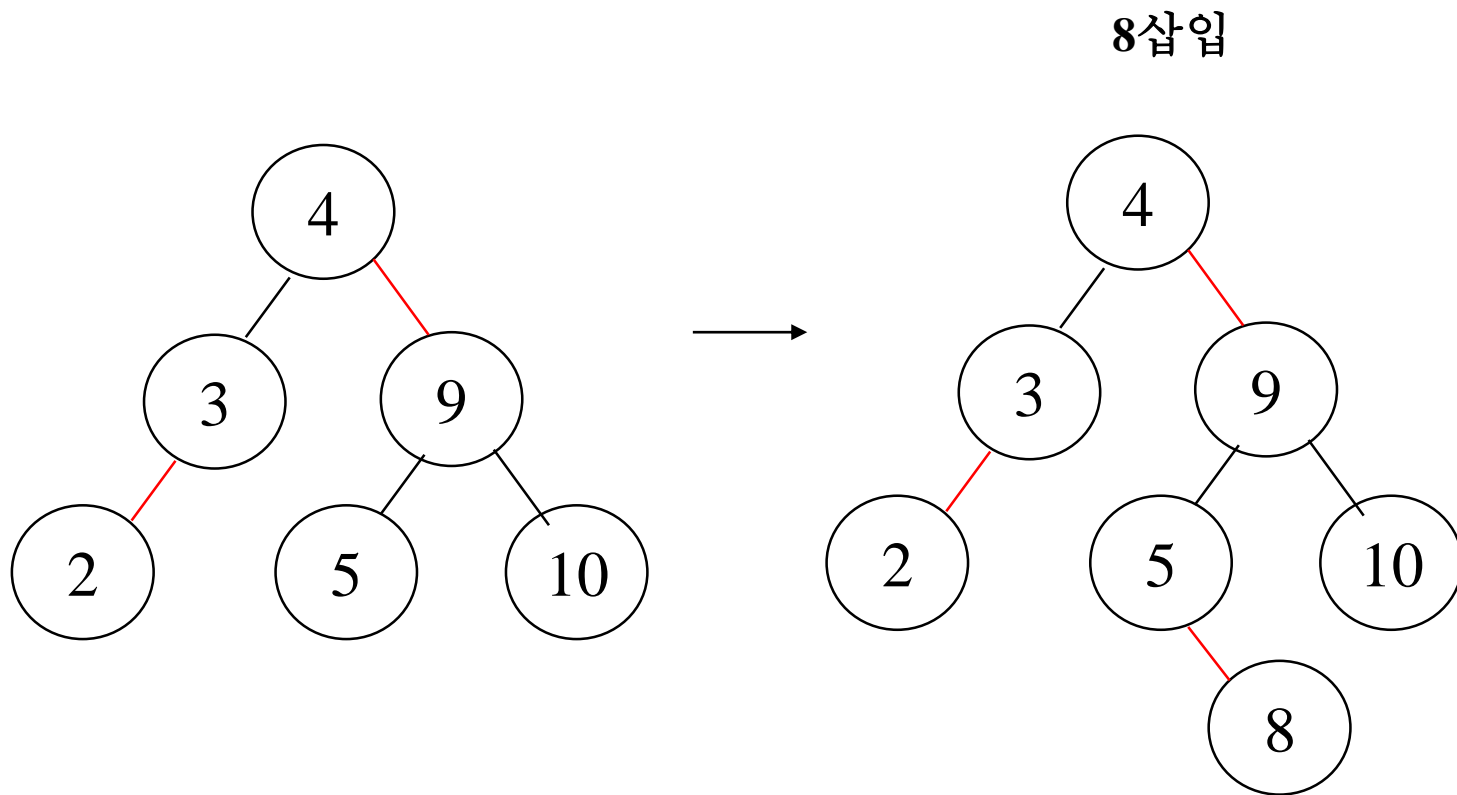
- ◆ 공백 레드 블랙 트리에서 키 5, 4, 3, 2, 10, 9, 8을 기술한 순서대로 삽입한 결과를 그려라. (Draw red-black tree after 5, 4, 3, 2, 10, 9, 8 inserted one by one.)

9삽입



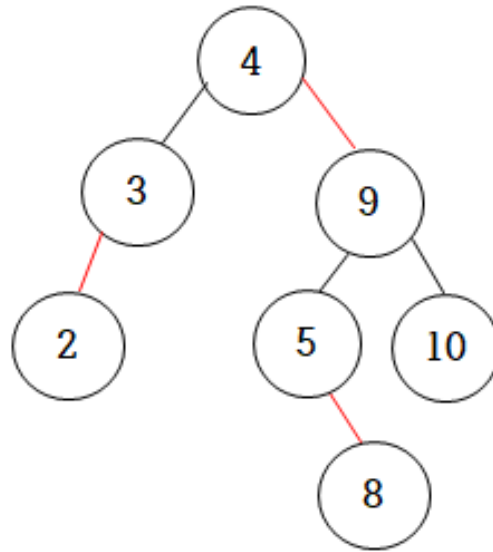
문제 14

- ◆ 공백 레드 블랙 트리에서 키 5, 4, 3, 2, 10, 9, 8을 기술한 순서대로 삽입한 결과를 그려라. (Draw red-black tree after 5, 4, 3, 2, 10, 9, 8 inserted one by one.)



문제 14

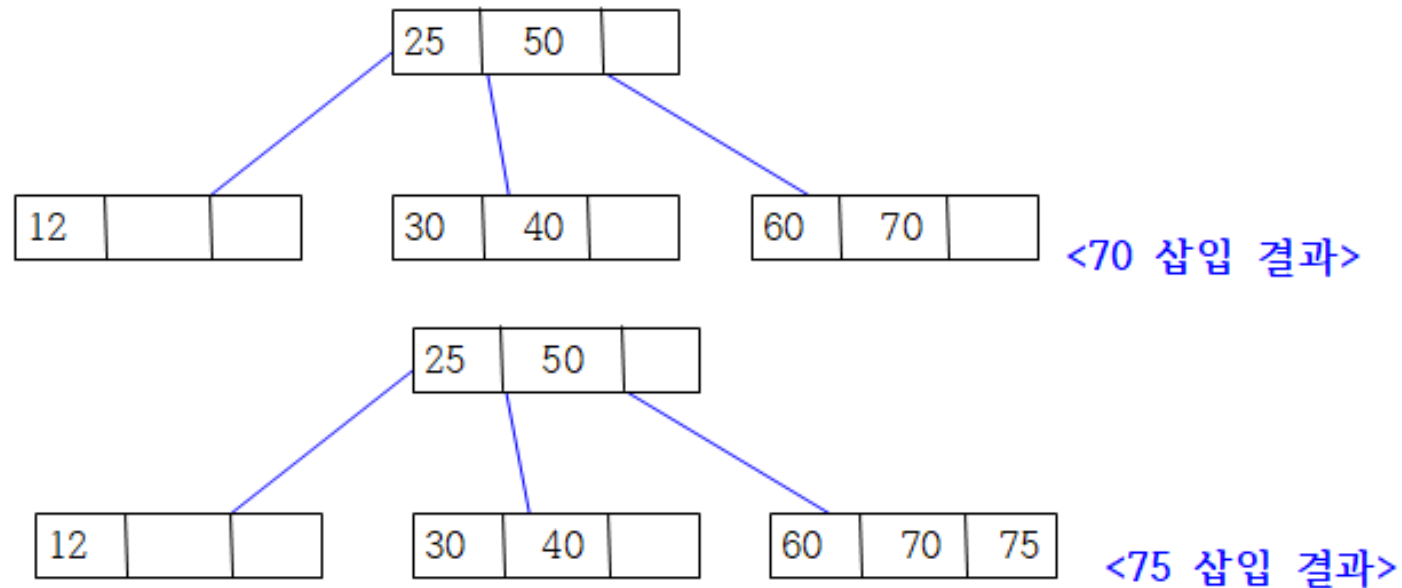
- ◆ 공백 레드 블랙 트리에서 키 5, 4, 3, 2, 10, 9, 8을 기술한 순서대로 삽입한 결과를 그려라. (Draw red-black tree after 5, 4, 3, 2, 10, 9, 8 inserted one by one.)



<삽입 결과>

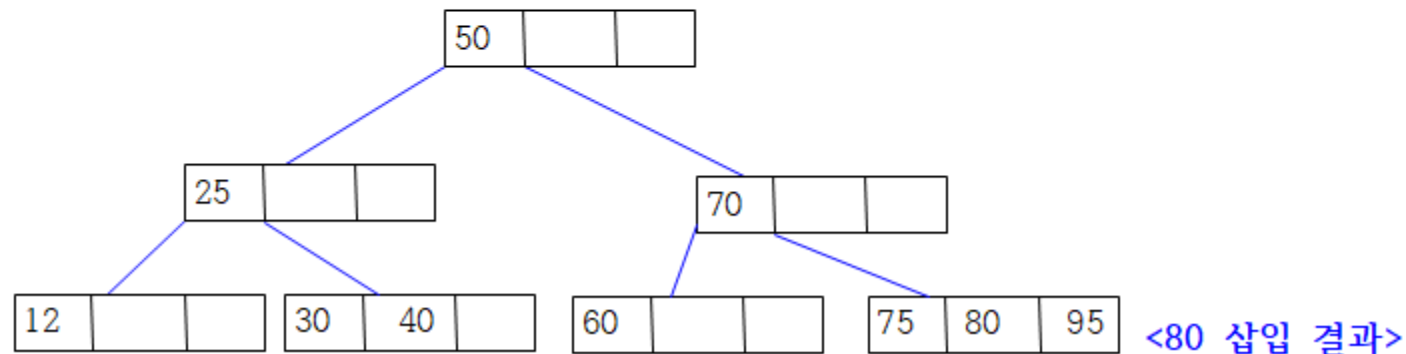
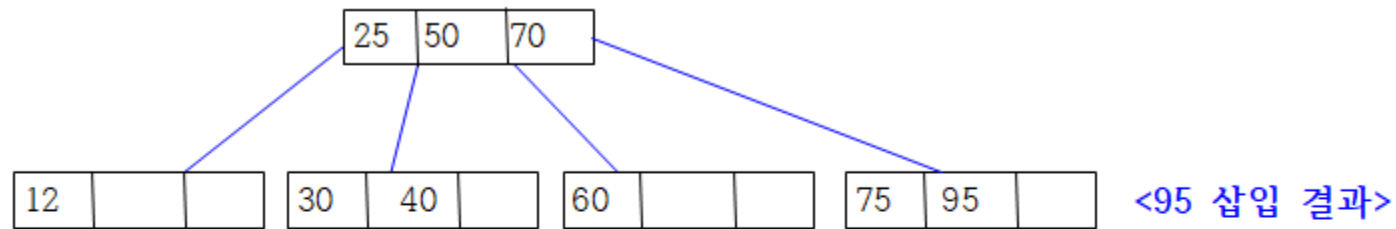
문제 15-1

- ◆ 다음 그림 2의 2-3-4 트리에 70, 75, 95, 80을 순서대로 삽입하라. 중간 과정과 그 결과를 보여라 (Draw 2-3-4 tree in Fig 2 after 70, 75, 95, 80 inserted one by one.)



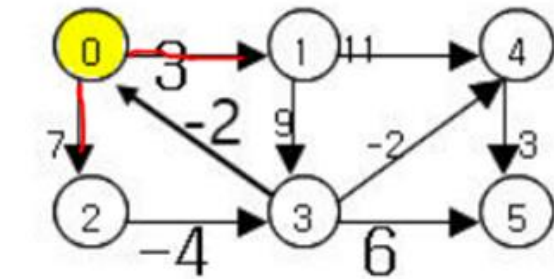
문제 15-2

- ◆ 다음 그림 2의 2-3-4 트리에 70, 75, 95, 80을 순서대로 삽입하라. 중간 과정과 그 결과를 보여라 (Draw 2-3-4 tree in Fig 2 after 70, 75, 95, 80 inserted one by one.)



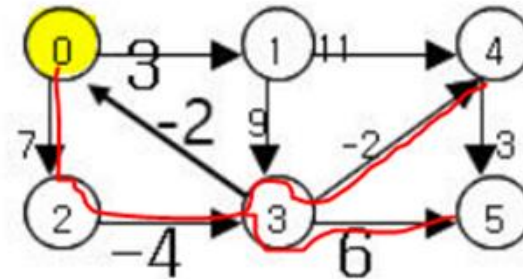
문제 17-1

- ◆ Bellman & Ford 알고리즘을 사용하여 그림 3의 출발점 0부터 나머지까지 최단거리를 구하라. 중간 과정을 적어라. (Find Shortest distances from 0 to other vertex using Bellman & Ford.)



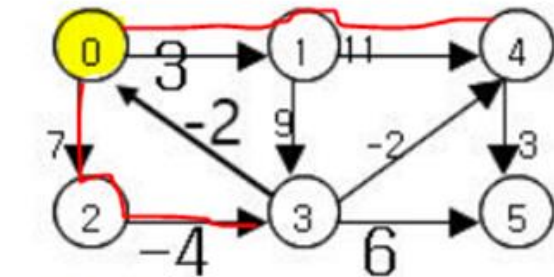
0	1	2	3	4	5
1	0	3	7	∞	∞

<간선 1개>



0	1	2	3	4	5
1	0	3	7	∞	∞
2	0	3	7	3	14
3	0	3	7	3	1

<간선 3개>

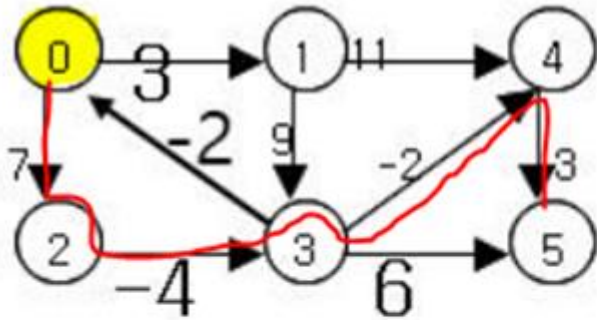


0	1	2	3	4	5
1	0	3	7	∞	∞
2	0	3	7	3	14

<간선 2개>

문제 17-2

- ◆ Bellman & Ford 알고리즘을 사용하여 그림 3의 출발점 0부터 나머지까지 최단거리를 구하라. 중간 과정을 적어라. (Find Shortest distances from 0 to other vertex using Bellman & Ford.)



	0	1	2	3	4	5
0	0	3	7	-2	3	10
1	∞	0	∞	9	11	∞
2	∞	∞	0	-4	3	∞
3	∞	∞	∞	0	-2	6
4	∞	∞	∞	∞	0	3
5	∞	∞	∞	∞	∞	0

<간선 4개>

<간선 5개> 변동 없음

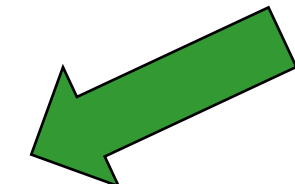
문제 18-1

- ◆ Floyd 알고리즘을 사용하여 그림 3의 모든 정점에서 모든 정점까지의 최단거리를 구하라. 중간 과정을 적어라. (Find shortest distances from all to all vertex using Floyd)


	0	1	2	3	4	5
0	0	3	7	∞	∞	∞
1	∞	0	∞	9	11	∞
2	∞	∞	0	-4	∞	∞
3	-2	1	5	0	-2	6
4	∞	∞	∞	∞	0	3
5	∞	∞	∞	∞	∞	0


<정점 1 경유>

	0	1	2	3	4	5
0	0	3	7	12	14	∞
1	∞	0	∞	9	11	∞
2	∞	∞	0	-4	∞	∞
3	-2	1	5	0	-2	6
4	∞	∞	∞	∞	0	3
5	∞	∞	∞	∞	∞	0


<정점 2 경유>

	0	1	2	3	4	5
0	0	3	7	3	14	∞
1	∞	0	∞	9	11	∞
2	∞	∞	0	-4	∞	∞
3	-2	1	5	0	-2	6
4	∞	∞	∞	∞	0	3
5	∞	∞	∞	∞	∞	0


<정점 3 경유>

	0	1	2	3	4	5
0	0	3	7	3	1	9
1	7	0	14	9	7	15
2	-6	-3	0	-4	-6	2
3	-2	1	5	0	-2	6
4	∞	∞	∞	∞	0	3
5	∞	∞	∞	∞	∞	0

문제 18-2

- ◆ Floyd 알고리즘을 사용하여 그림 3의 모든 정점에서 모든 정점까지의 최단거리를 구하라. 중간 과정을 적어라. (Find shortest distances from all to all vertex using Floyd)

	0	1	2	3	4	5
0	0	3	7	3	1	4
1	7	0	14	9	7	10
2	-6	-3	0	-4	-6	-3
3	-2	1	5	0	-2	1
4	∞	∞	∞	∞	0	3
5	∞	∞	∞	∞	∞	0

<정점 4 경유>

<정점 5 경유> 변동 없음

	0	1	2	3	4	5
0	0	3	7	3	1	4
1	7	0	14	9	7	10
2	-6	-3	0	-4	-6	-3
3	-2	1	5	0	-2	1
4	∞	∞	∞	∞	0	3
5	∞	∞	∞	∞	∞	0

<결과>

2017 문제 19

레드 블랙 트리의 탐색 시간 복잡도를 적어라

답: $O(\log_2 n)$

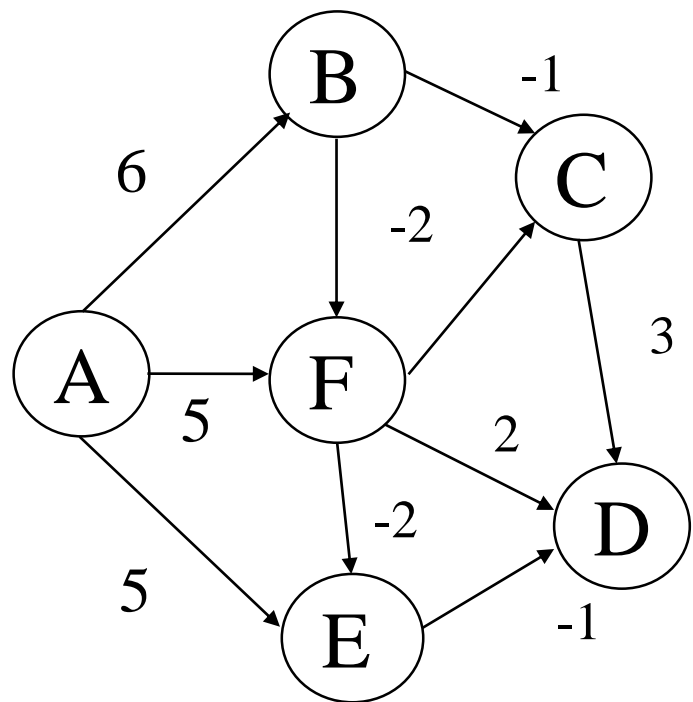
문제 20

키를 비교하여 정렬하는 알고리즘 중 가장 빠른
알고리즘은(A)정렬 알고리즘이며,
기수 정렬 알고리즘의 시간 복잡도는 (B)이다.

답->A:Quick B: $O(n)$

문제 21

그림6에서 0으로부터 다른 정점간의 최단거리를
bellman&ford 알고리즘을 이용하여 구하시오

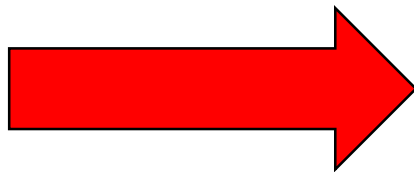
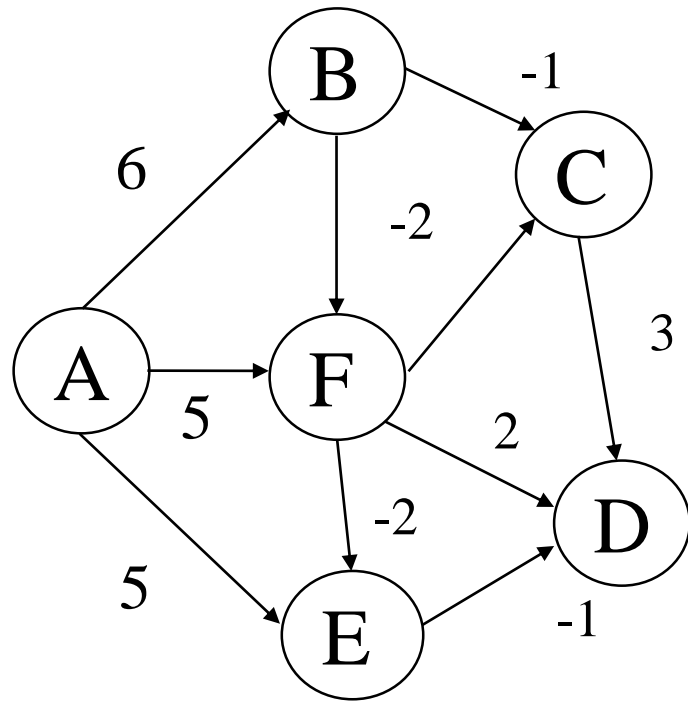


	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞
1	0	6	∞	∞	5	5
2	0	6	5	4	3	4
3	0	6	5	2	2	4
4	0	6	5	1	2	4
5	0	6	5	1	2	4

문제 22

그림6에서 모든 정점에서부터 모든 정점까지의
최단 경로를 구하시오.

floyd



	a	b	c	d	e	f
a	0	6	∞	∞	5	5
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0

정점A 경우

	a	b	c	d	e	f
a	0	6	∞	∞	5	5
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0



정점B 경우

	a	b	c	d	e	f
a	0	6	5	∞	5	4
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0



정점 c 경우

	a	b	c	d	e	f
a	0	6	5	8	5	4
b	∞	0	-1	2	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0



정점 D 경우

	a	b	c	d	e	f
a	0	6	5	8	5	4
b	∞	0	-1	2	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0



정점 E 경유

	a	b	c	d	e	f
a	0	6	5	4	5	4
b	∞	0	-1	2	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	-3	-2	0



정점 F 경유

	a	b	c	d	e	f
a	0	6	5	1	2	4
b	∞	0	-1	-5	-4	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	-3	-2	0

최단경로

2020 문제 19

레드 블랙 트리가 2-3-4 트리와 비교하여 가지는
장점을 설명하시오.

1. 레드 블랙트리는 이진 탐색트리의 함수를 거의 그대로 사용하여 노드 구조와 삽입삭제 코드가 2-3-4 트리보다 간결하다
2. 2-3-4트리와 마찬가지로 단일 패스 삽입 삭제가 적용된다.
3. 언제 회전에 균형을 잡아야 하는지 쉽게 판별된다.

문제 20

2-3-4트리가 2-3트리와 비교하여
가지는 장점을 설명하라

1. 삽입 삭제시 한번의 패스만 발생하고, 루트로 영향을 주지 않는다.
2. 후진부할 일어나지 않는다

2017 문제 19

레드 블랙 트리의 탐색 시간 복잡도를 적어라

답: $O(\log_2 n)$

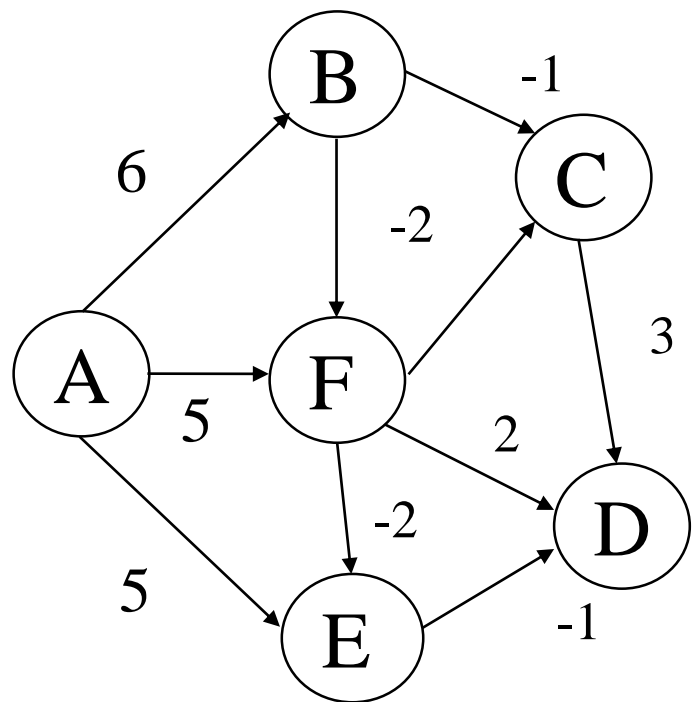
문제 20

키를 비교하여 정렬하는 알고리즘 중 가장 빠른
알고리즘은(A)정렬 알고리즘이며,
기수 정렬 알고리즘의 시간 복잡도는 (B)이다.

답->A:Quick B: $O(n)$

문제 21

그림6에서 0으로부터 다른 정점간의 최단거리를
bellman&ford 알고리즘을 이용하여 구하시오

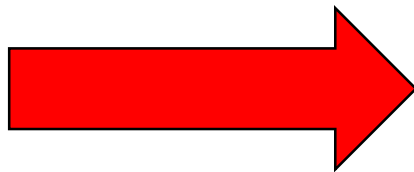
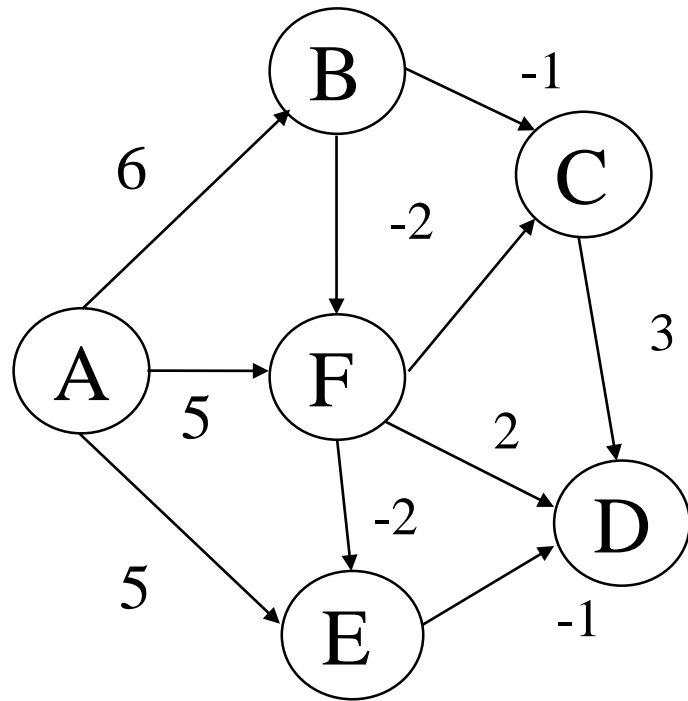


	a	b	c	d	e	f
0	0	∞	∞	∞	∞	∞
1	0	6	∞	∞	5	5
2	0	6	5	4	3	4
3	0	6	5	2	2	4
4	0	6	5	1	2	4
5	0	6	5	1	2	4

문제 22

그림6에서 모든 정점에서부터 모든 정점까지의
최단 경로를 구하시오.

floyd



	a	b	c	d	e	f
a	0	6	∞	∞	5	5
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0

정점A 경우

	a	b	c	d	e	f
a	0	6	∞	∞	5	5
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0



정점B 경우

	a	b	c	d	e	f
a	0	6	5	∞	5	4
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0



정점 c 경우

	a	b	c	d	e	f
a	0	6	5	8	5	4
b	∞	0	-1	2	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0



정점 D 경우

	a	b	c	d	e	f
a	0	6	5	8	5	4
b	∞	0	-1	2	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0



정점 E 경유

	a	b	c	d	e	f
a	0	6	5	4	5	4
b	∞	0	-1	2	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	-3	-2	0



정점 F 경유

	a	b	c	d	e	f
a	0	6	5	1	2	4
b	∞	0	-1	-5	-4	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	-3	-2	0

→ 최단경로

2020 문제 19

레드 블랙 트리가 2-3-4 트리와 비교하여 가지는
장점을 설명하시오.

1. 레드 블랙트리는 이진 탐색트리의 함수를 거의 그대로 사용하여 노드 구조와 삽입삭제 코드가 2-3-4 트리보다 간결하다
2. 2-3-4트리와 마찬가지로 단일 패스 삽입 삭제가 적용된다.
3. 언제 회전에 균형잡아야 하는지 쉽게 판별된다.
4. 가장 빠른 탐색 알고리즘이다.

문제 20

2-3-4트리가 2-3트리와 비교하여
가지는 장점을 설명하라

1. 삽입 삭제시 한번의 패스만 발생하고, 루트로 영향을 주지 않는다.
2. 후진부할 일어나지 않는다

<레드 블랙 트리가 2-3-4 트리와 비교하여 가지는 장점을 설명하라.>

- ◆ 레드 블랙 트리 : 모든 노드가 빨강과 검정으로 되어있고, 빨강과 검정 노드는 규칙에 따라 삽입 삭제 rotation 된다.
- ◆ 2-3-4 트리를 “이진 트리”로 표현한 것 이다.

<장점>

- ◆ 2-3-4 트리는 노드 구조가 복잡하고 삽입,삭제 코드가 복잡하다.
 - ◆ 레드 블랙 트리는 이진 탐색 트리의 함수를 거의 그대로 사용하며 2-3-4 트리의 장점인 단일 패스 삽입,삭제가 그대로 적용된다.
 - ◆ 언제 회전에 의해 균형을 잡아야 하는지 쉽게 판별할 수 있다.
 - ◆ 레드 블랙 트리는 탐색 시간이 가장 빠르다.
-
- ◆ 따라서 레드 블랙 트리는 2-3-4 트리의 장점을 가지고 단점인 자식수가 많은 것을 2개로 제한한 트리이다.

<2-3-4 트리가 2-3 트리와 비교하여 가지는 장점을 설명하라.>

- ♦ 2-3-4 트리 : 3개의 키와 4개의 자식을 가질 수 있고, 2-3 트리와는 다르게 내려가며 노드를 분할한다.

<장점>

- ♦ 2-3트리는 후진 분할이 일어나지만, 2-3-4트리는 후진분할이 일어나지 않아 삽입,삭제 시 root에 영향을 주지 않는다.

<레드블랙 트리의 탐색 시간 복잡도를 적어라>

- ◆ 이진 탐색 트리에서 탐색, 삽입, 삭제의 시간 복잡도는 $O(h)$ 이다. 즉, 트리의 높이에 비례한 만큼의 시간이 걸린다.
- ◆ 레드 블랙 트리는 이진탐색 트리로, 여러 조건을 통해 시간 복잡도를 $O(\log n)$ 으로 줄일 수 있다.
- ◆ 정답 : $O(\log n)$

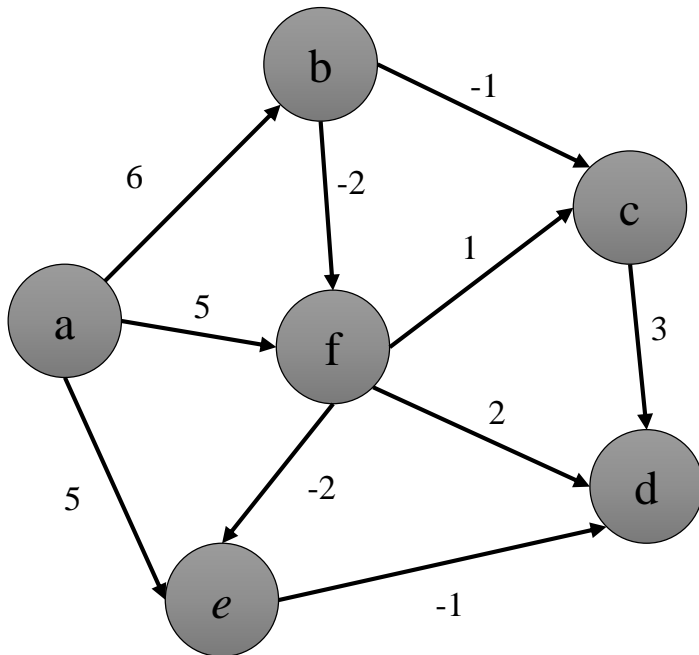
2017-2 기말고사 문제 20번

- ◆ 키를 비교하여 정렬하는 알고리즘 중 가장 빠른 알고리즘은 (Quick) 정렬 알고리즘이며, 기수 정렬 알고리즘의 시간 복잡도는 ($O(M*n)$) 이다. (M: 자릿수 , n: 원소 갯수)
- ◆ 키를 비교하여 정렬하는 알고리즘은 $n \log n$ 보다 빠를 수 없다.
 - ◆ Insertion sort
 - ◆ Selection sort
 - ◆ Bubble sort
 - ◆ Quick sort
 - ◆ Merge sort
 - ◆ Heap sort
- ◆ 정답 : **Quick, $O(M * n)$**

<그림 6에서 0으로부터 다른 정점간의 최단거리를 bellman&ford 알고리즘을 이용하여 구하라.>

Bellman & ford 알고리즘

- 갈수 있는 간선의 수를 늘려가며 최단 거리를 구한다. $O(v^2)$ 시간이 걸린다.



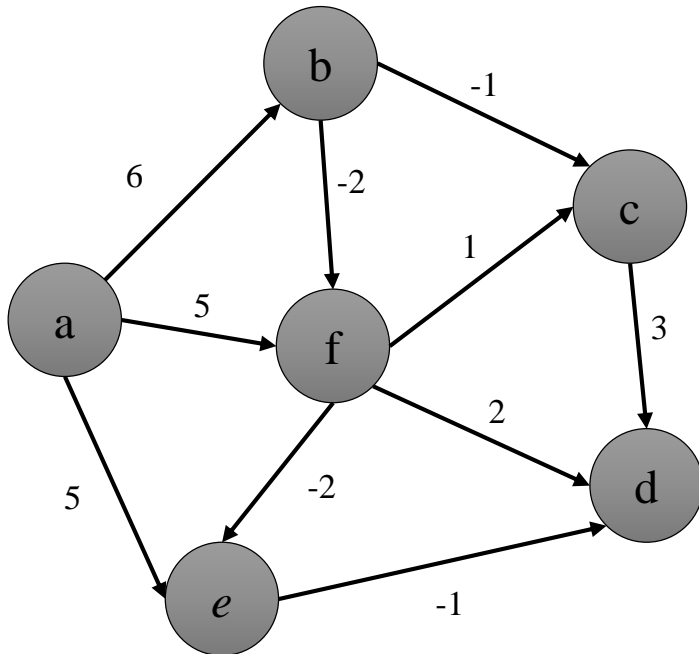
0(a)에서 시작

	a	b	c	d	e	f
Dist0	0	∞	∞	∞	∞	∞
Dist1	0	6	∞	∞	5	5
Dist2	0	6	5	4	3	4
Dist3	0	6	5	2	2	4
Dist4	0	6	5	1	2	4
Dist5	0	6	5	1	2	4

<그림 6에서 모든 정점에서부터 모든 정점까지의 최단 경로를 구하라.>

Floyd Wallshall 알고리즘

- 여러 정점에서 시작해서 모든 정점까지의 최단 경로를 구하는 알고리즘
- 행렬로 구하고 $O(v^3)$ 시간이 걸림



1. $D(a) = a$ 를 거치는 경우 최단경로 변화 없음

	a	b	c	d	e	f
a	0	6	∞	∞	5	5
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0

<그림 6에서 모든 정점에서부터 모든 정점까지의 최단 경로를 구하라.>

2. $D(b) = b$ 를 거치는 경우 최단경로

	a	b	c	d	e	f
a	0	6	5	∞	5	4
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0

3. $D(c) = c$ 를 거치는 경우 최단경로

	a	b	c	d	e	f
a	0	6	5	8	5	4
b	∞	0	-1	2	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	2	-2	0

<그림 6에서 모든 정점에서부터 모든 정점까지의 최단 경로를 구하라.>

4. $D(d) = d$ 를 거치는 경우 최단경로 변화 없음

5. $D(e) = e$ 를 거치는 경우 최단경로

	a	b	c	d	e	f
a	0	6	5	4	5	4
b	∞	0	-1	∞	∞	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	-3	-2	0

6. $D(f) = f$ 를 거치는 경우 최단경로

	a	b	c	d	e	f
a	0	6	5	1	2	4
b	∞	0	-1	-5	-4	-2
c	∞	∞	0	3	∞	∞
d	∞	∞	∞	0	∞	∞
e	∞	∞	∞	-1	0	∞
f	∞	∞	1	-3	-2	0