

# 완전 탐색 2

최백준 [choi@startlink.io](mailto:choi@startlink.io)

---

# 일부 경우만 해보기

---

# 일부 경우만 해보기

Two Pointer Algorithm

3

- 모든 경우 다해보기와 다르게 절대 정답이 될 수 없는 경우는 확인하지도 않을 수 있다

# 수들의 합 2

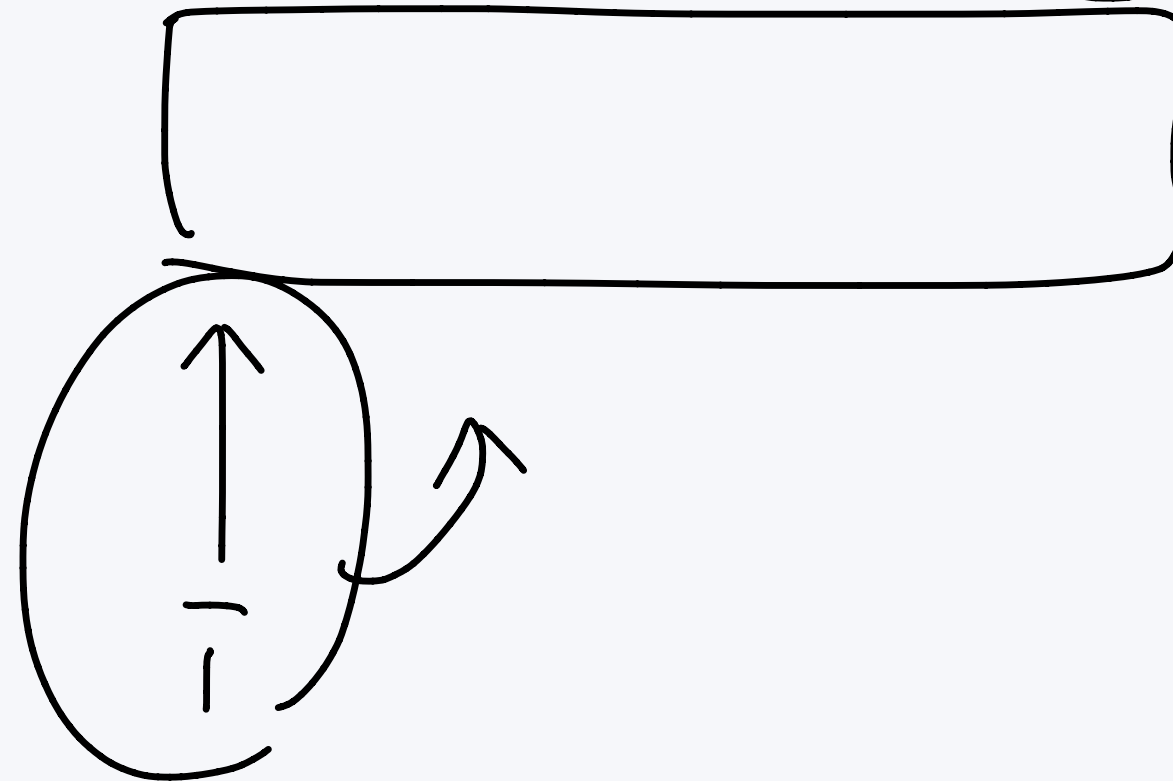
<https://www.acmicpc.net/problem/2003>

- N개의 수로 된 수열  $A[1], A[2], \dots, A[N]$  이 있다
- 이 수열의 i번째 수부터 j번째 수까지의 합  $A[i] + A[i+1] + \dots + A[j-1] + A[j]$ 가 M이 되는 경우의 수를 구하는 문제

$O(N)$

$O(N^2)$

j



$O(N)$

$< M$  : 구간  $\uparrow$   $j++$

$> M$  : 구간  $\downarrow$   $i++$

# 수들의 합 2

<https://www.acmicpc.net/problem/2003>

- $i$ 와  $j$ 가 있을 때, 그 사이의 합이  $M$ 보다 크면 합이 작아져야 하기 때문에  $i$ 를 증가시키고
- 합이  $M$ 보다 작으면 합이 커져야 하기 때문에  $j$ 를 증가시킨다

# 수들의 합 2

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 1

구간:  $L \sim R$

$1 < 5$



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



# 수들의 합 2

7

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 3



# 수들의 합 2

8

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 6

구간 ↓





# 수들의 합 2

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 5 (찾았다!)



# 수들의 합 2

10

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 9



# 수들의 합 2

11

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 7



# 수들의 합 2

12

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 4



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

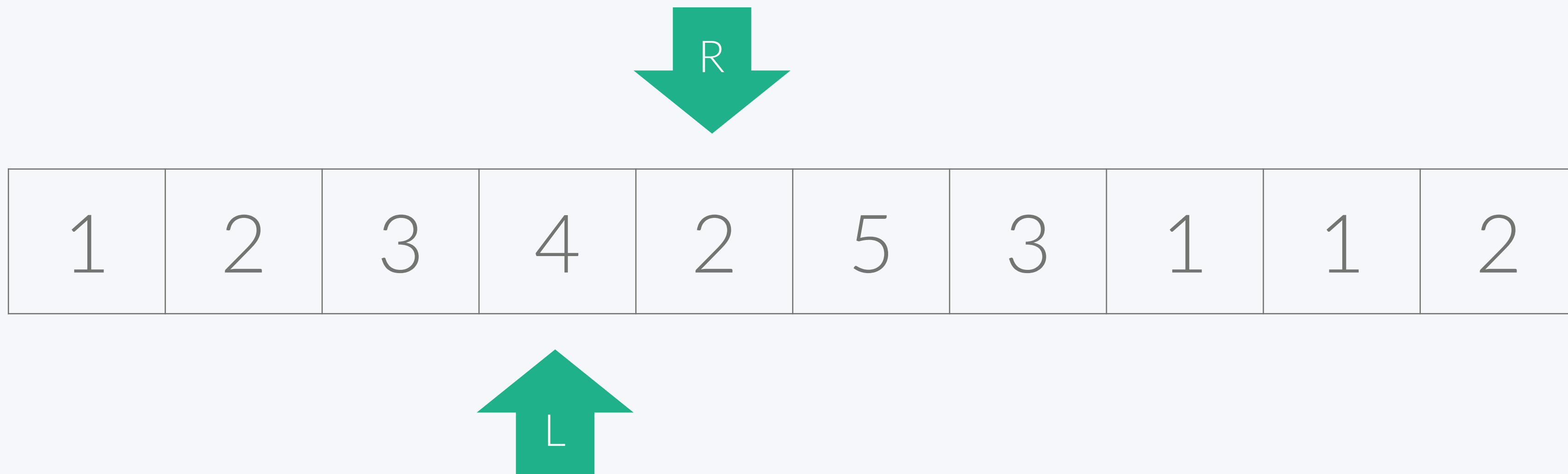


# 수들의 합 2

13

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 6

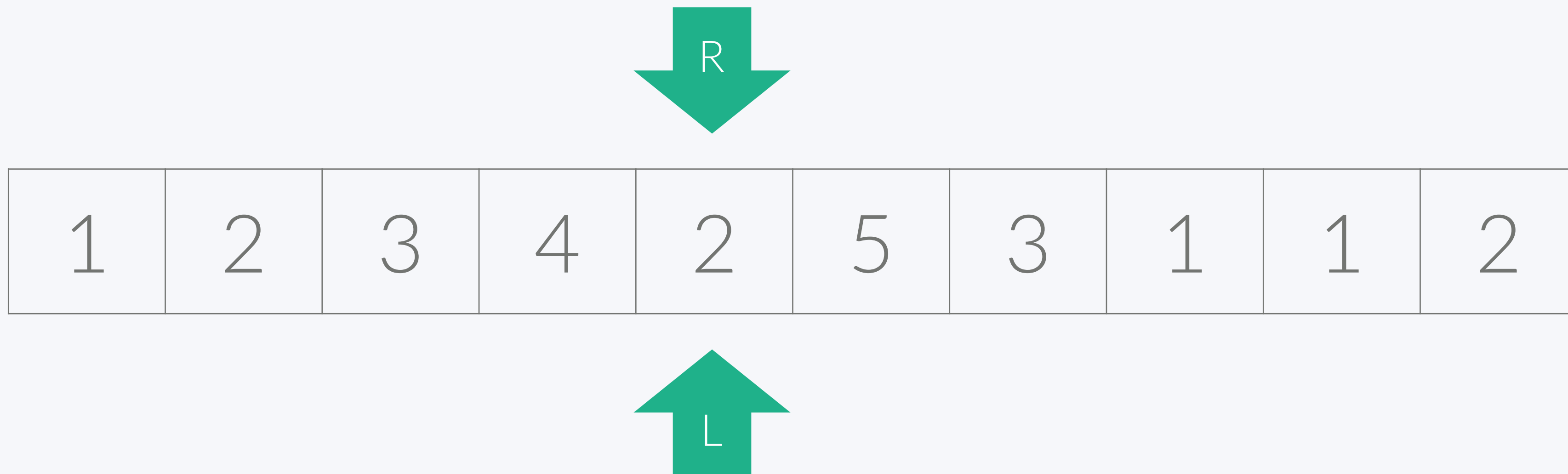


# 수들의 합 2

14

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 2

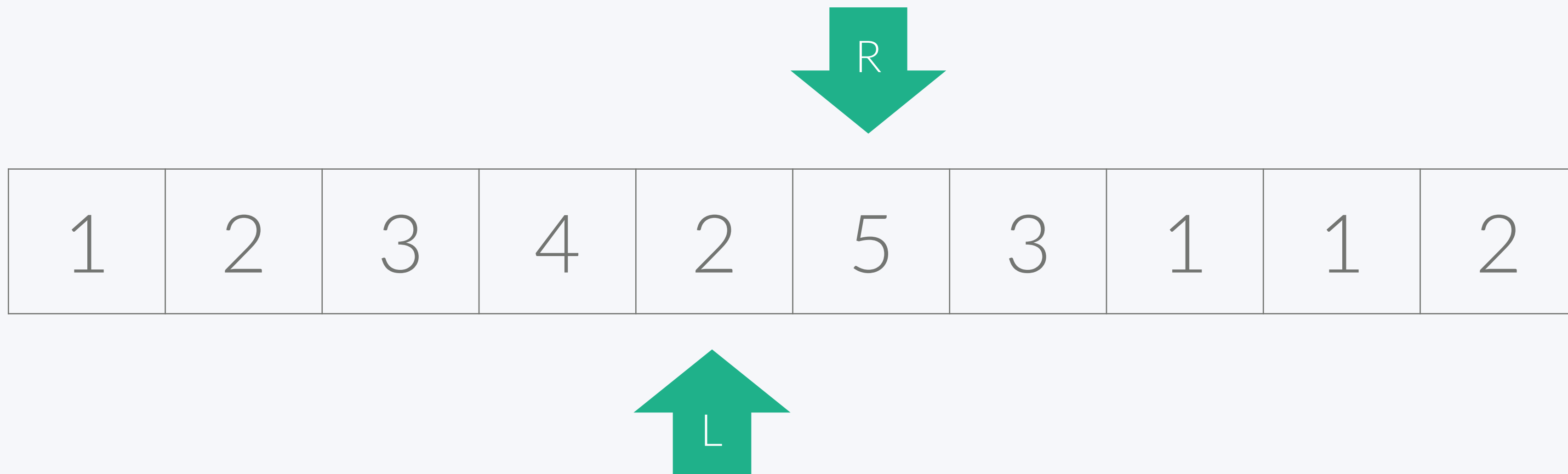


# 수들의 합 2

15

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 7



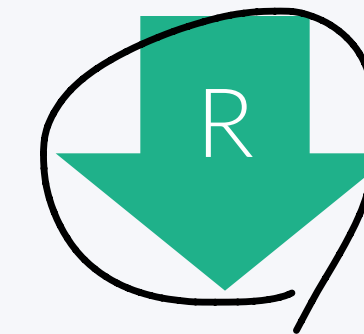
# 수들의 합 2

$$L \leq R$$

16

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 5 (찾았다!)
- 같은 경우에는 L, R 둘 중에 아무거나 증가해도 상관없지만
- 이런 경우 때문에 R이 증가해야 한다.



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



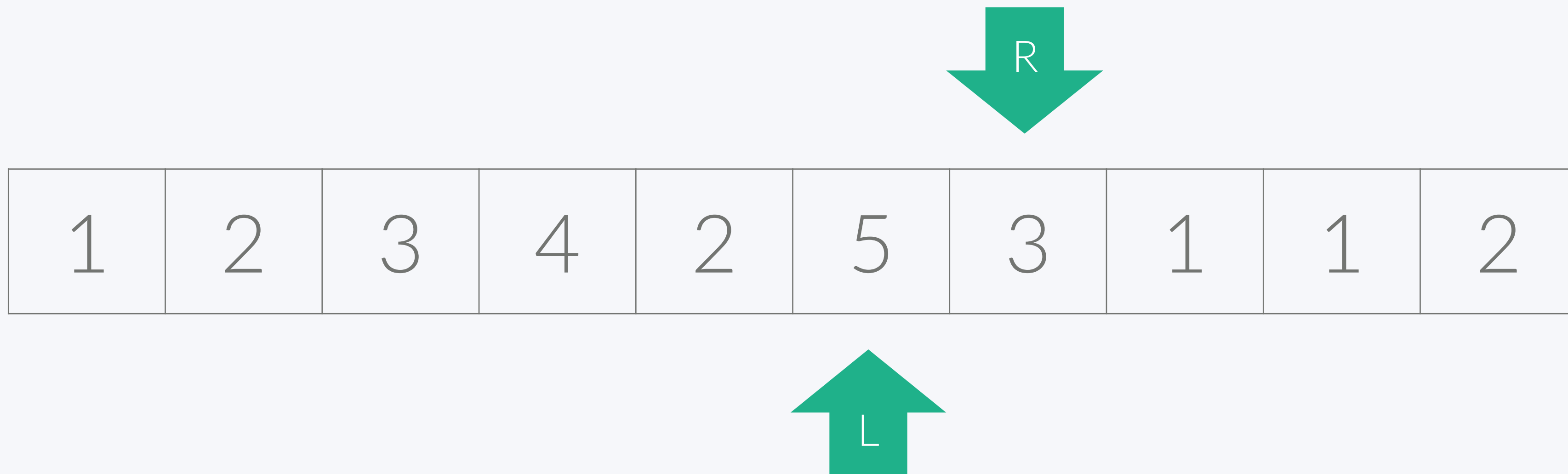


# 수들의 합 2

17

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 8

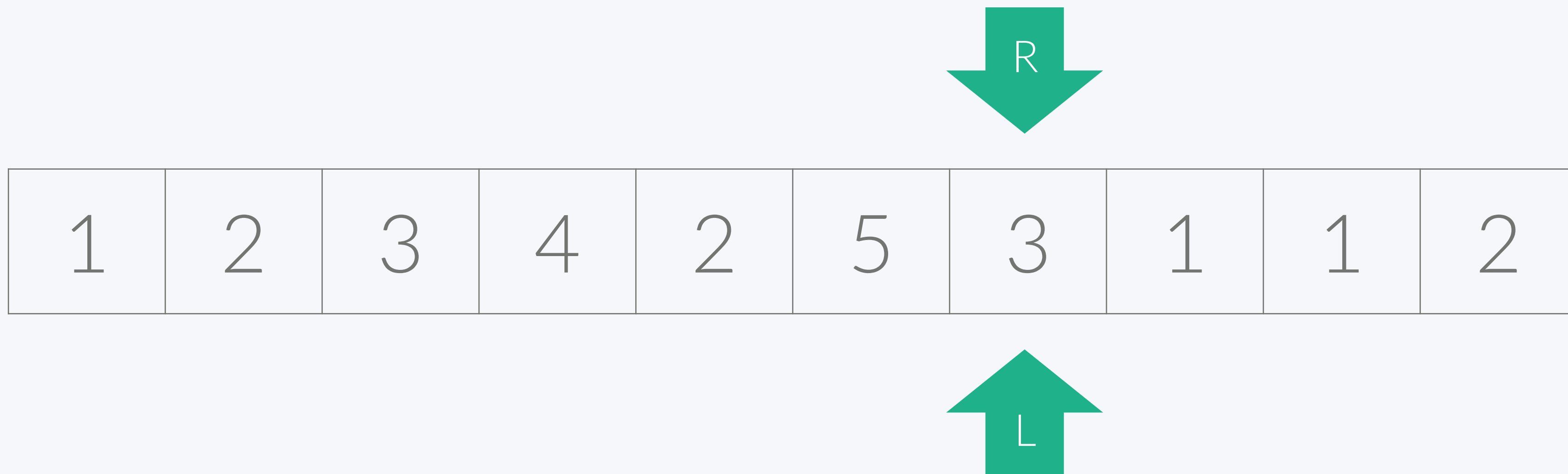


# 수들의 합 2

18

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 3



# 수들의 합 2

19

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 4



# 수들의 합 2

20

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 5 (찾았다!)

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



# 수들의 합 2

21

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 7

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



# 수들의 합 2

22

<https://www.acmicpc.net/problem/2003>

- 찾으려고 하는 수: 5
- 합: 4

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



# 수들의 합 2

<https://www.acmicpc.net/problem/2003>

23

- 찾으려고 하는 수: 5
- 끝

$$\textcircled{R} \geq N$$



# 수들의 합 2

<https://www.acmicpc.net/problem/2003>

```
int left=0, right=0, sum=a[0], ans = 0;
```

```
while (left <= right && right < n) {
```

```
    if (sum < m) {
```

```
        right += 1;
```

```
        sum += a[right]; ←
```

```
    } else if (sum == m) {
```

```
        ans += 1;
```

```
        right += 1; ←
```

```
        sum += a[right];
```

```
    } else if (sum > m) {
```

```
        sum -= a[left];
```

```
        left++;
```

```
    }
```

```
}
```

Sum: left ~ right  
→

→  $O(N)$



# 수들의 합 2

<https://www.acmicpc.net/problem/2003>

- C/C++: <https://gist.github.com/Baekjoon/9cac4b8e2d4e0310dec2>
- Java: <https://gist.github.com/Baekjoon/886927f47c13809f58a2>

# 부분합

<https://www.acmicpc.net/problem/1806>

- 구간 합 중에서 합이  $S$  이상인 것 중에서 가장 짧은 것을 구하는 문제

# 부분합

<https://www.acmicpc.net/problem/1806>

- C/C++: <https://gist.github.com/Baekjoon/9ef1d516e28e46285cfb>
- Java: <https://gist.github.com/Baekjoon/907e123eeee200f1eebc>

# 소수의 연속합

<https://www.acmicpc.net/problem/1644>

- 수들의 합 2 문제와 같지만, 소수를 구해서 답을 구해야 하는 문제

(N)

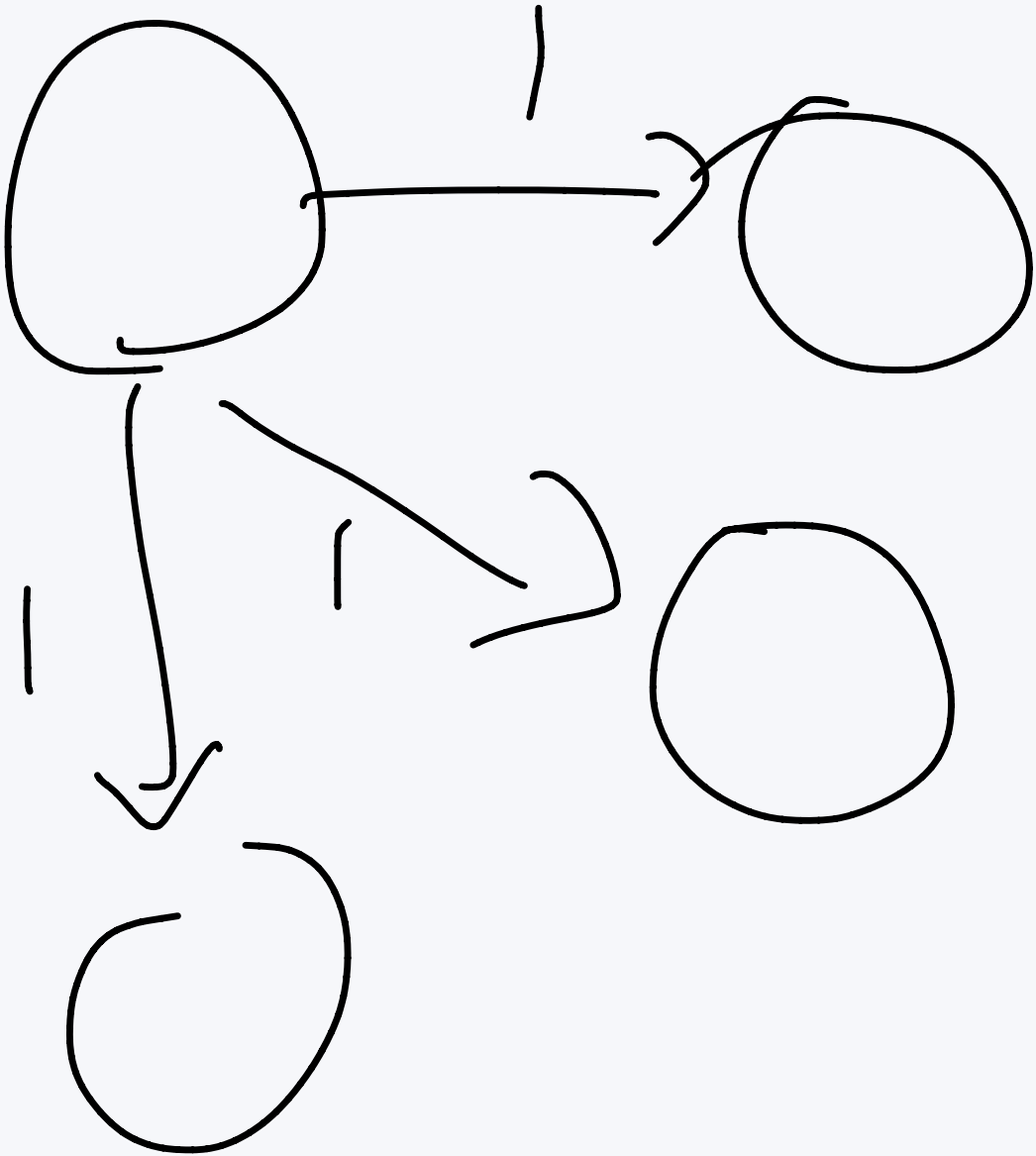
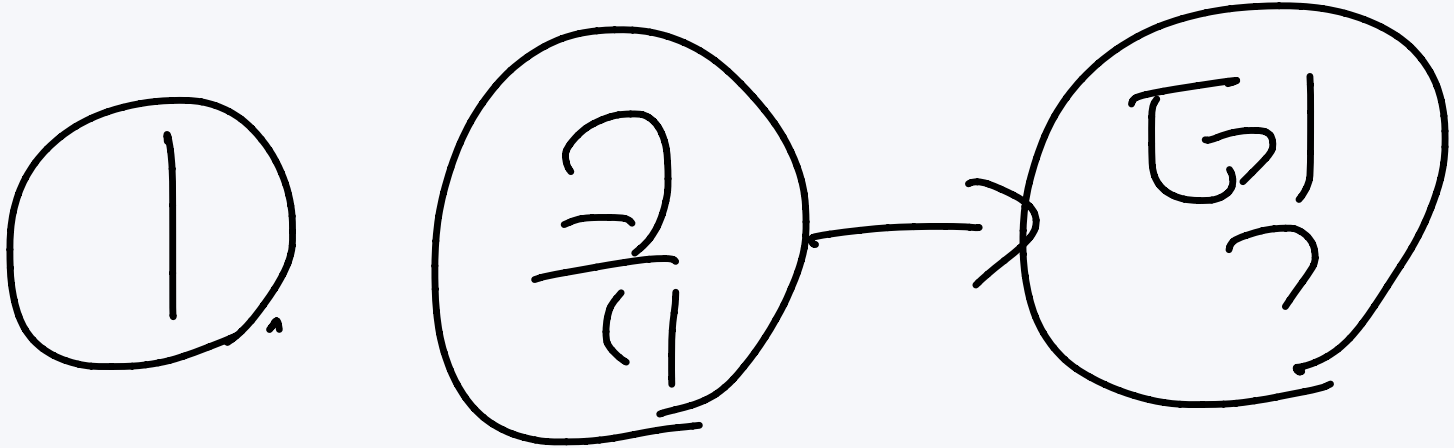
|   |   |   |   |    |    |    |
|---|---|---|---|----|----|----|
| 2 | 3 | 5 | 7 | 11 | 13 | 17 |
|---|---|---|---|----|----|----|

# 소수의 연속합

<https://www.acmicpc.net/problem/1644>

- C: <https://gist.github.com/Baekjoon/99f99c0c3b3281488d60>
- C++: <https://gist.github.com/Baekjoon/d5dd94afcd4414b09f37>
- Java: <https://gist.github.com/Baekjoon/02482585f719e2ece210>

# 덱 사용하기



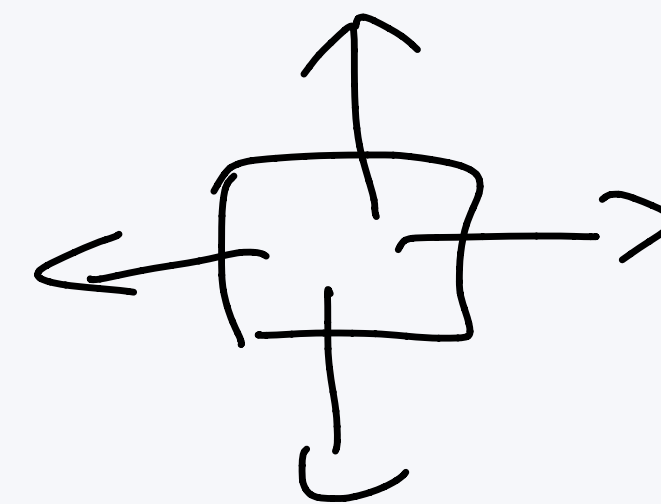
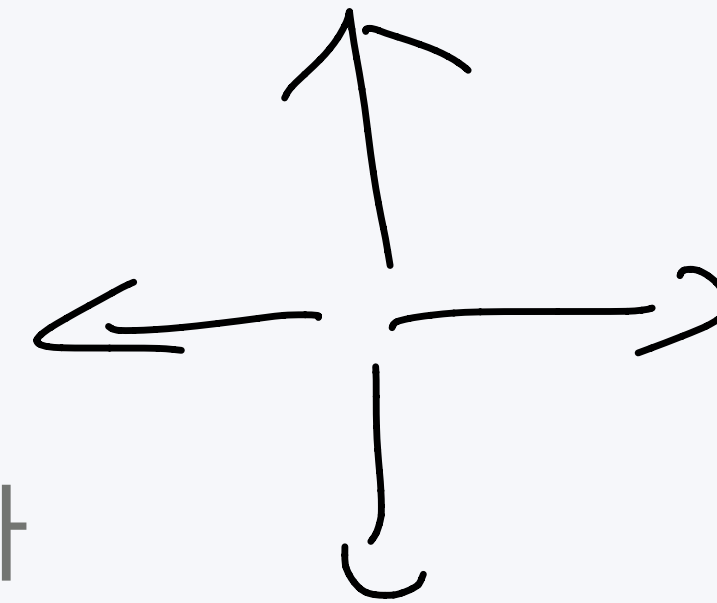
가중치 : 0 또는 1

# 알고스팟

<https://www.acmicpc.net/problem/1261>

- 미로는  $N \times M$  크기이고, 총  $1 \times 1$  크기의 방으로 이루어져 있다
- 빈 방은 자유롭게 다닐 수 있지만, 벽은 부수지 않으면 이동할 수 없다
- $(x, y)$ 에 있을 때, 이동할 수 있는 방은  $(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$  이다
- $(1, 1)$ 에서  $(N, M)$ 으로 이동하려면 벽을 최소 몇 개 부수어야 하는지 구하는 문제

벽을 부수는 방의 최소 개수



- 처음 상태

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |

[illegible]



# 알고스팟

<https://www.acmicpc.net/problem/1261>

- 벽을 부수지 않고 이동할 수 있는 곳

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |

|   |   |  |  |  |  |
|---|---|--|--|--|--|
| 0 | 0 |  |  |  |  |
| 0 |   |  |  |  |  |
| 0 | 0 |  |  |  |  |
|   |   |  |  |  |  |
|   |   |  |  |  |  |
|   |   |  |  |  |  |

# 알고스팟

<https://www.acmicpc.net/problem/1261>

- 벽을 1개 부수고 이동할 수 있는 곳

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |

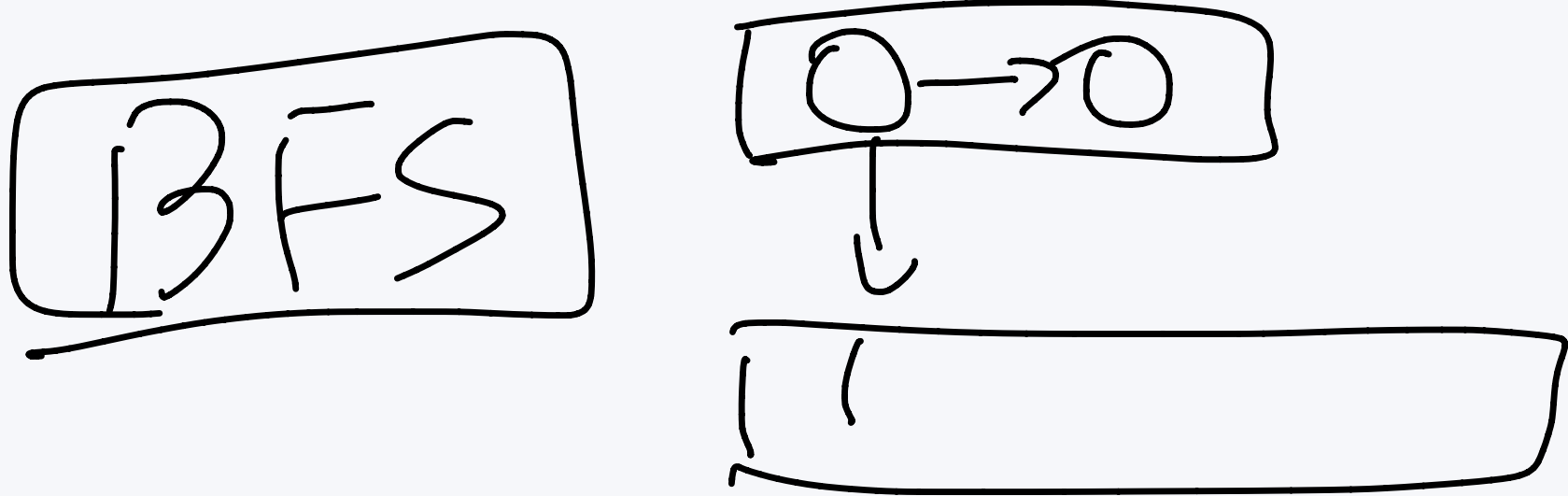
|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 1 |   |   |   |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 |   |   |   |
| 1 | 1 | 1 | 1 | 1 |   |
| 1 |   | 1 | 1 |   |   |
|   | 1 | 1 | 1 |   |   |

# 알고스팟

<https://www.acmicpc.net/problem/1261>

- 벽을 2개 부수고 이동할 수 있는 곳

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |



|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 2 | 2 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 2 | 2 | 2 |
| 1 | 1 | 1 | 1 | 1 | 2 |
| 1 | 2 | 1 | 1 | 2 | 2 |
| 2 | 1 | 1 | 1 | 2 | 2 |

# 알고스팟

<https://www.acmicpc.net/problem/1261>

- BFS탐색을 벽을 부순 횟수에 따라서 나누어서 수행해야 한다.
- C/C++: <https://gist.github.com/Baekjoon/3d8ed2a3976c7affbd73>
- Java: <https://gist.github.com/Baekjoon/e66c8bed6b5d440d5bafbfe882acc19e>
- 시간 복잡도:  $O(N^2)$

# 알고스팟

37

<https://www.acmicpc.net/problem/1261>

- 우선 순위 큐를 사용해서, **벽을 뚫은 횟수**가 적은 것부터 처리하도록 구현할 수 있다.
- C/C++: <https://gist.github.com/Baekjoon/e6f554163ab8f8e7b25f>
- 시간 복잡도:  $O(N^2 \lg N)$



# 알고스팟

<https://www.acmicpc.net/problem/1261>

38

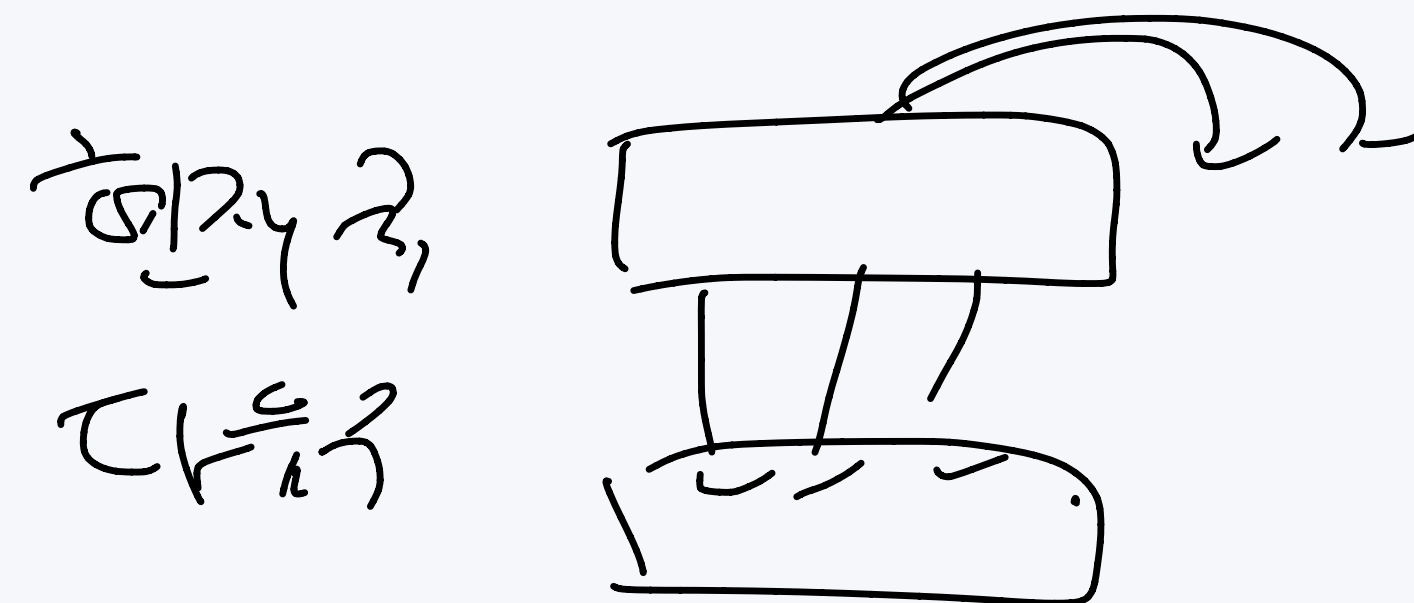
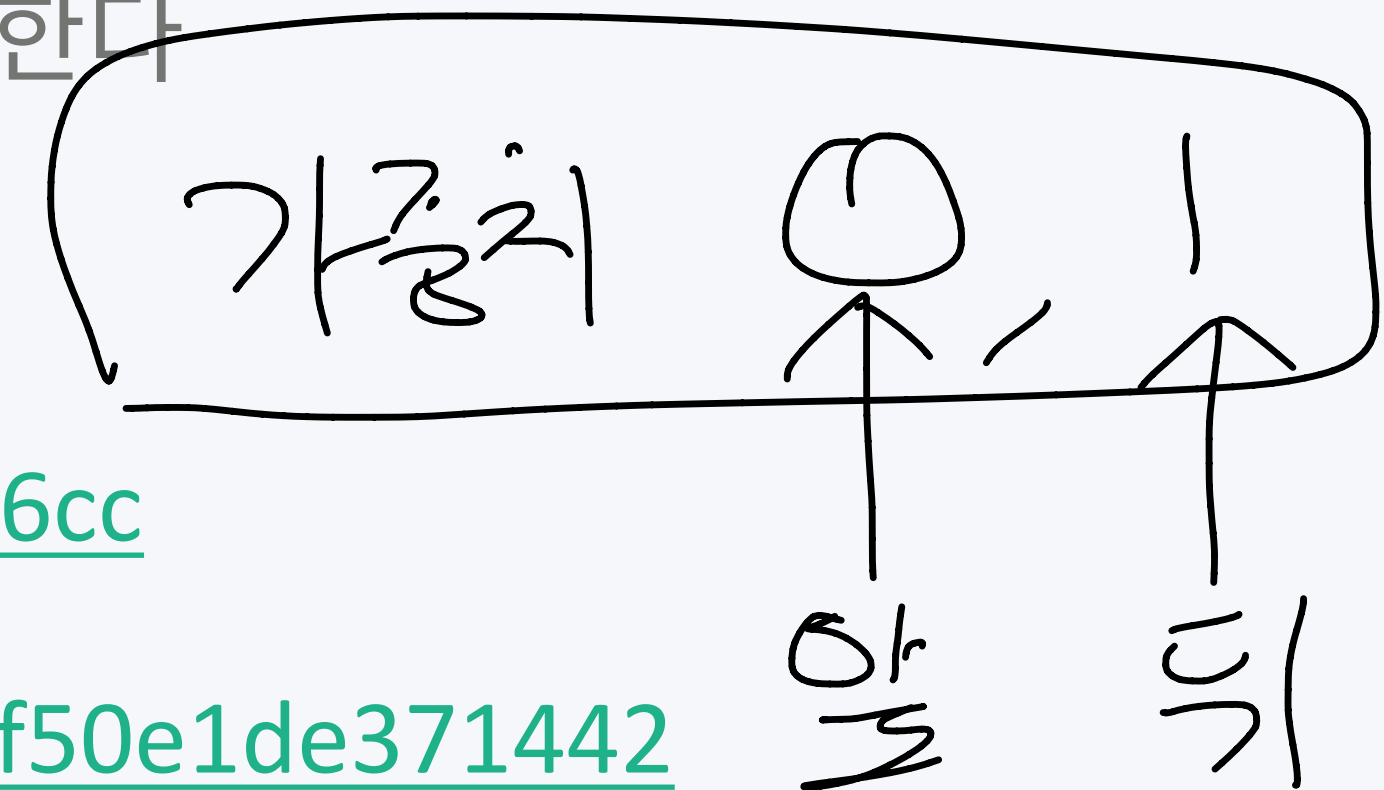
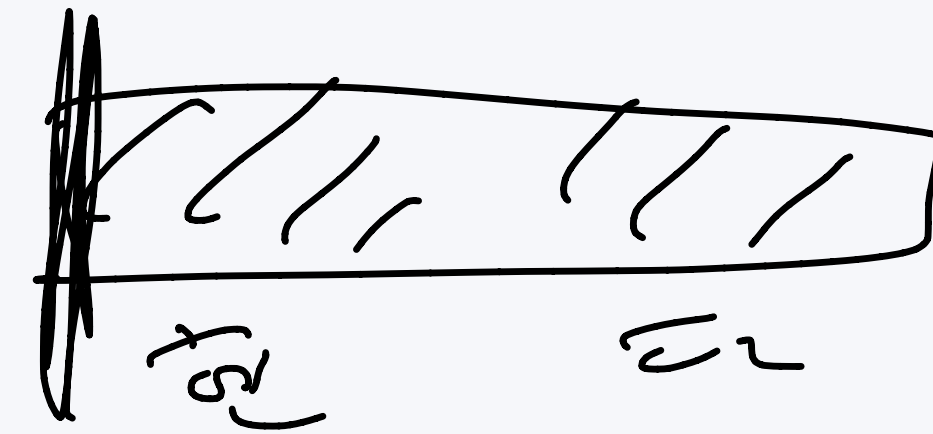
- 어차피 벽을 뚫는다고 안 뚫는다고 나누어지기 때문에, 덱을 사용한다

- 벽을 뚫는 경우에는 뒤에, 안 뚫는 경우에는 앞에 추가한다.

- C/C++: <https://gist.github.com/Baekjoon/9da1eed82383645026cc>

- Java: <https://gist.github.com/Baekjoon/0cc05c42a1ab3b3b325f50e1de371442>

- 시간 복잡도:  $O(N^2)$



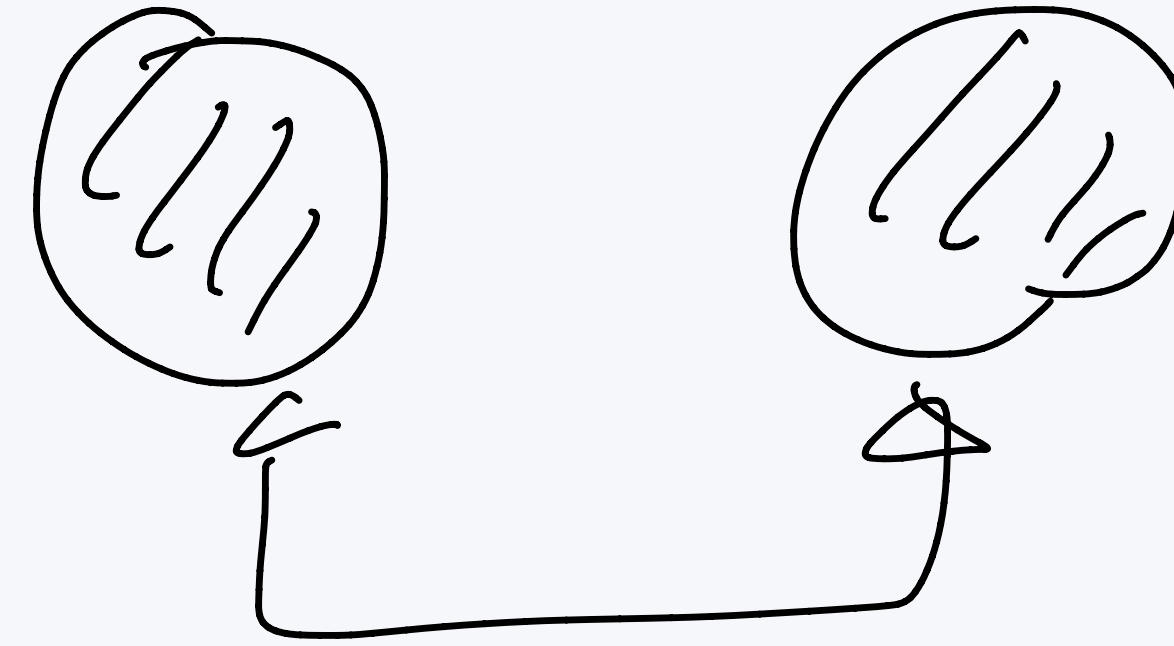
# 중간에서 만나기

---

# 중간에서 만나기

Meet in the Middle

- 문제를 절반으로 나눠서
- 양쪽 절반에서 모든 경우를 다 해보는 방법이다.
- 탐색의 크기가 많이 줄어든다.
- 문제의 크기가  $N$ 인 경우에  $2^N$  에서
- $M = N/2$  라고 했을 때,  $2^M + 2^M$  으로 줄어들게 된다.





# 부분집합의 합 2

41

<https://www.acmicpc.net/problem/1208>

하드마스크  
저귀호호

- 부분집합의 합 문제와 비슷하지만, 배열을 2개 써야하는 문제

- $A = [1, 2, 1, 3, 2, 1]$ ,  $M = 4$  인 경우를 생각해보자

- $A$ 를 절반으로 나누어서

- $Up = [1, 2, 1]$

- $Down = [3, 2, 1]$

- 에 대해서 각각 모든 경우를 나열한다.

크기  $N =$

$[1, 2, 1, 3, 2, 1]$

부분집합의 합

$M$

$N \leq 40$

$2^N$

$2^{20} = 1048576$

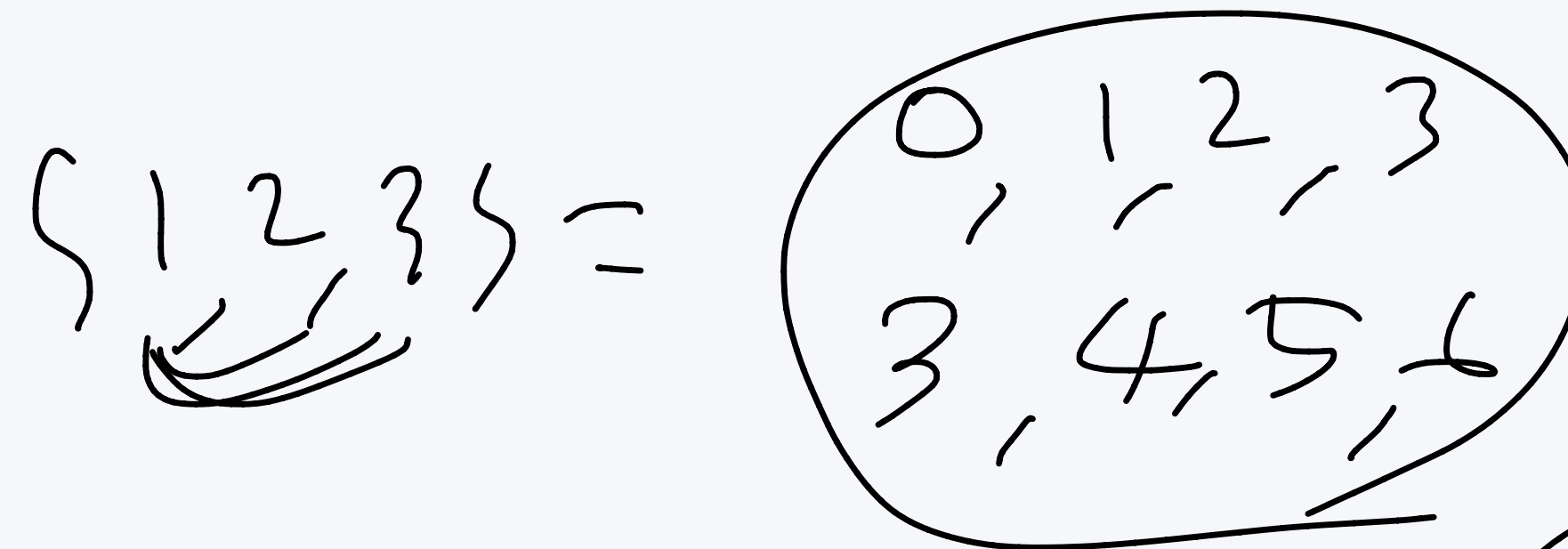
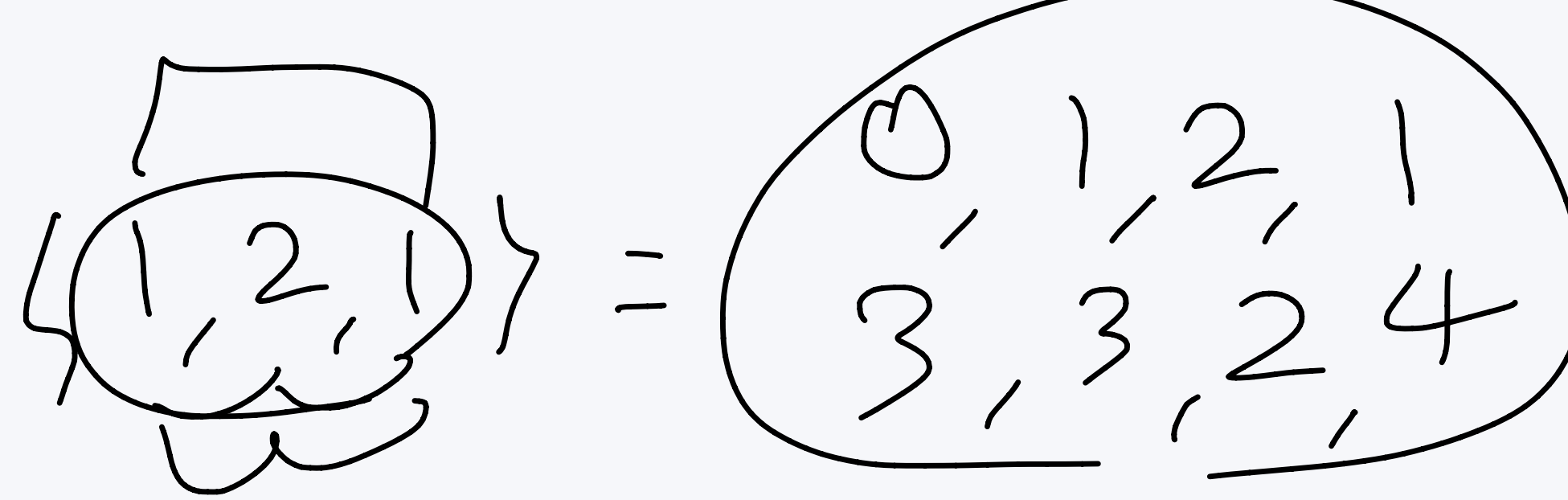
$2^{40} = [2^{20}]$

# 부분집합의 합 2

<https://www.acmicpc.net/problem/1208>

42

- Up = [1, 2, 1]
- Down = [1, 2, 3]
- 에 대해서 각각 모든 경우를 나열한다.



$M=4$

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 |
| 0 | 1 | 2 | 3 | 3 | 4 | 5 | 6 |

# 부분집합의 합 2

43

<https://www.acmicpc.net/problem/1208>

- $0 + 6 = 6$ 이고, 찾으려고 하는 수 4보다 크기 때문에, R을 1칸 당긴다.



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 |
|---|---|---|---|---|---|---|---|

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|



# 부분집합의 합 2

44

<https://www.acmicpc.net/problem/1208>

- $0 + 5 = 5$ 이고, 찾으려고 하는 수 4보다 크기 때문에, R을 1칸 당긴다.

Binary Search

$$0 + 4 = 4$$
$$1 + ? = 4$$

3

$$1 + 2 + 2 + 1$$



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 |
|---|---|---|---|---|---|---|---|

Down ~~B~~ =

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|



# 부분집합의 합 2

45

<https://www.acmicpc.net/problem/1208>

$$1 \times 1 = 1$$

- $0 + 4 = 4$ 이고, 찾으려고 하는 수 4 이다.
- 위에 0이 1개, 아래에 4가 1개 있기 때문에, 4는 총  $1 \times 1 = 1$ 개이다.
- 이제, L과 R을 이동시킨다.



|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 |
|---|---|---|---|---|---|---|---|

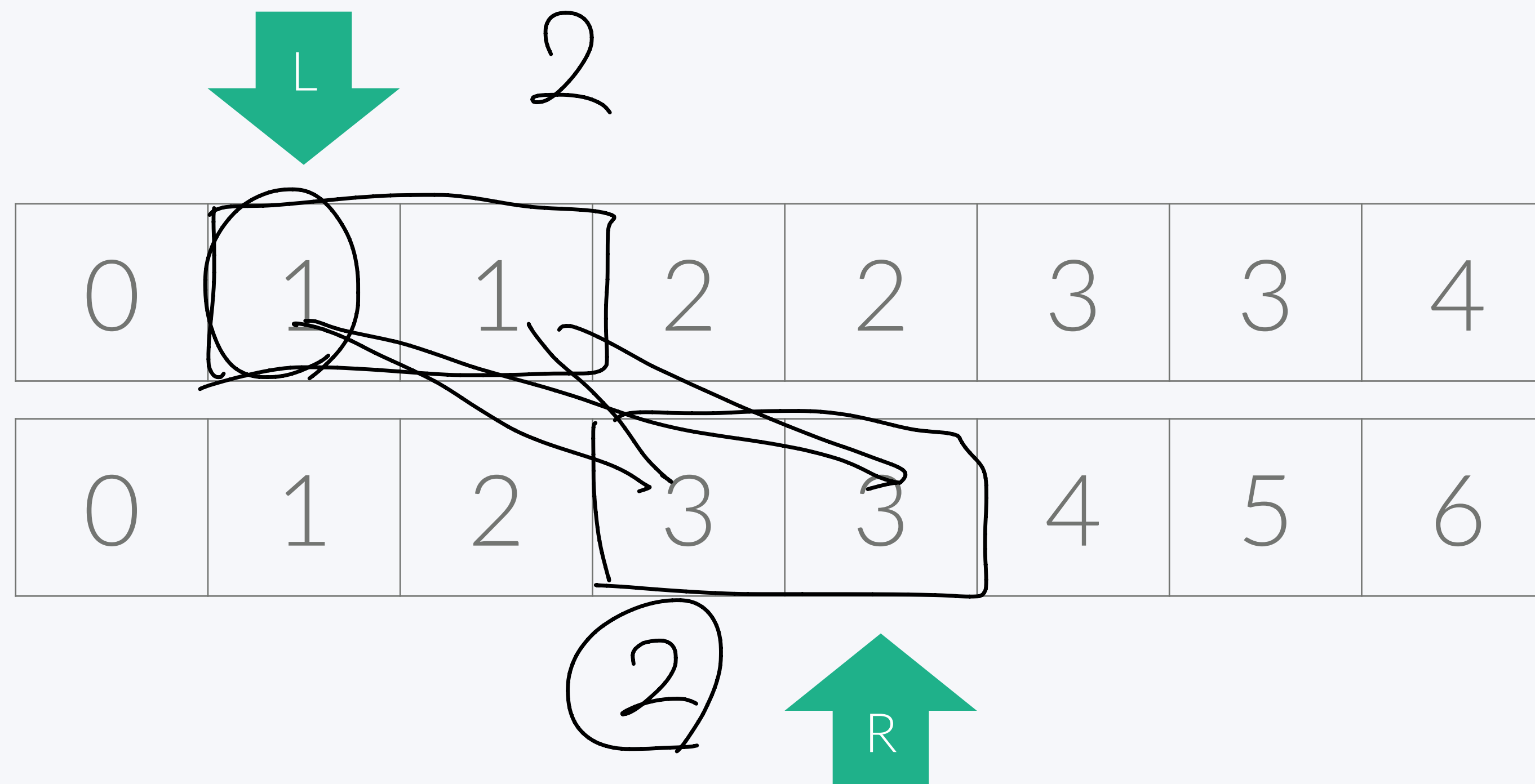
|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|



# 부분집합의 합 2

<https://www.acmicpc.net/problem/1208>

- $1 + 3 = 4$ 이고, 찾으려고 하는 수 4 이다.
- 위에 1이 2개, 아래에 3이 2개 있기 때문에, 4는 총  $2 \times 2 = 4$ 개이다.
- 이제, L과 R을 이동시킨다.



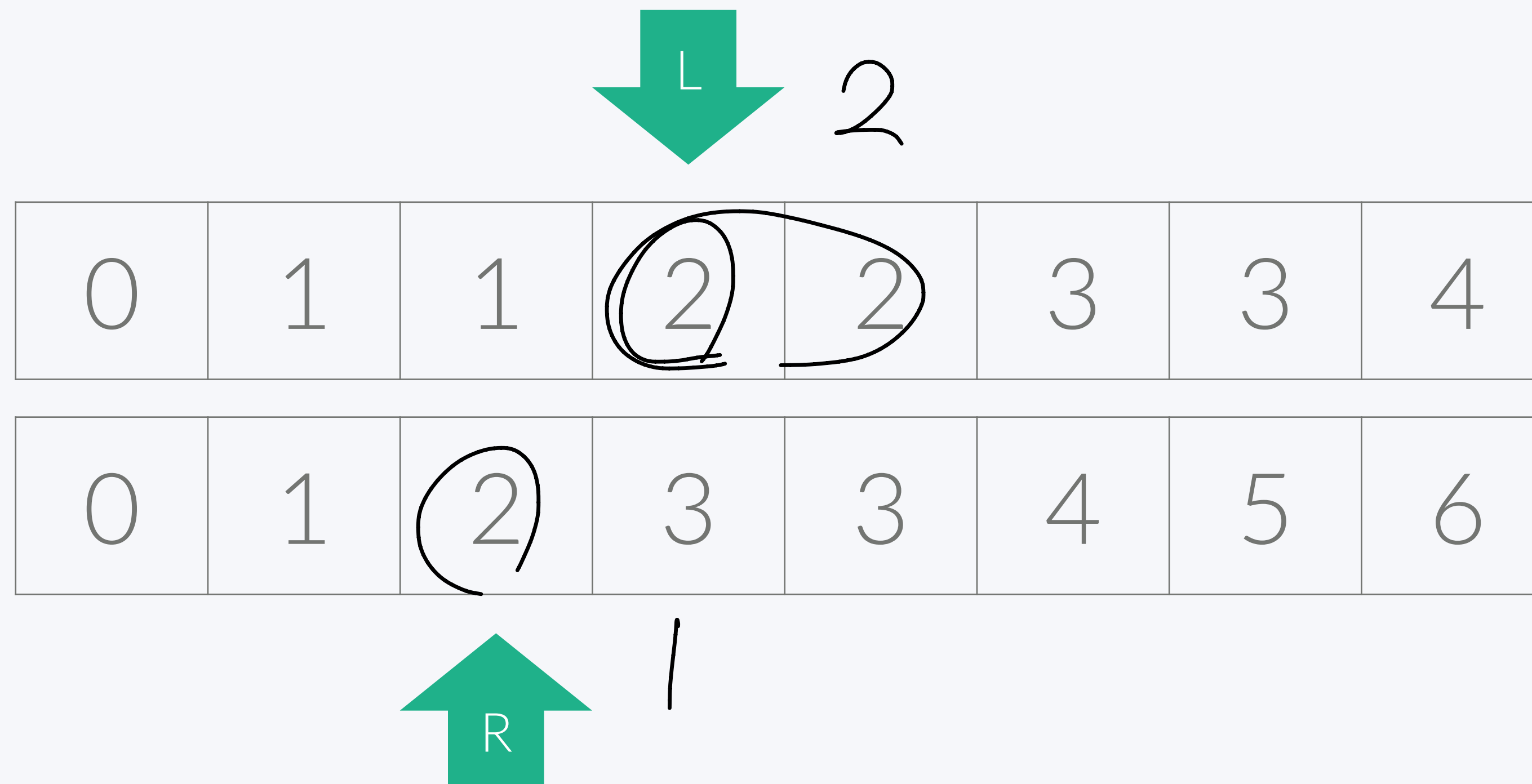
# 부분집합의 합 2

47

<https://www.acmicpc.net/problem/1208>

- $2 + 2 = 4$ 이고, 찾으려고 하는 수 4이다.
- 위에 2가 2개, 아래에 2가 1개 있기 때문에, 4는 총  $2 \times 1 = 2$ 개이다.
- 이제, L과 R을 이동시킨다.

2

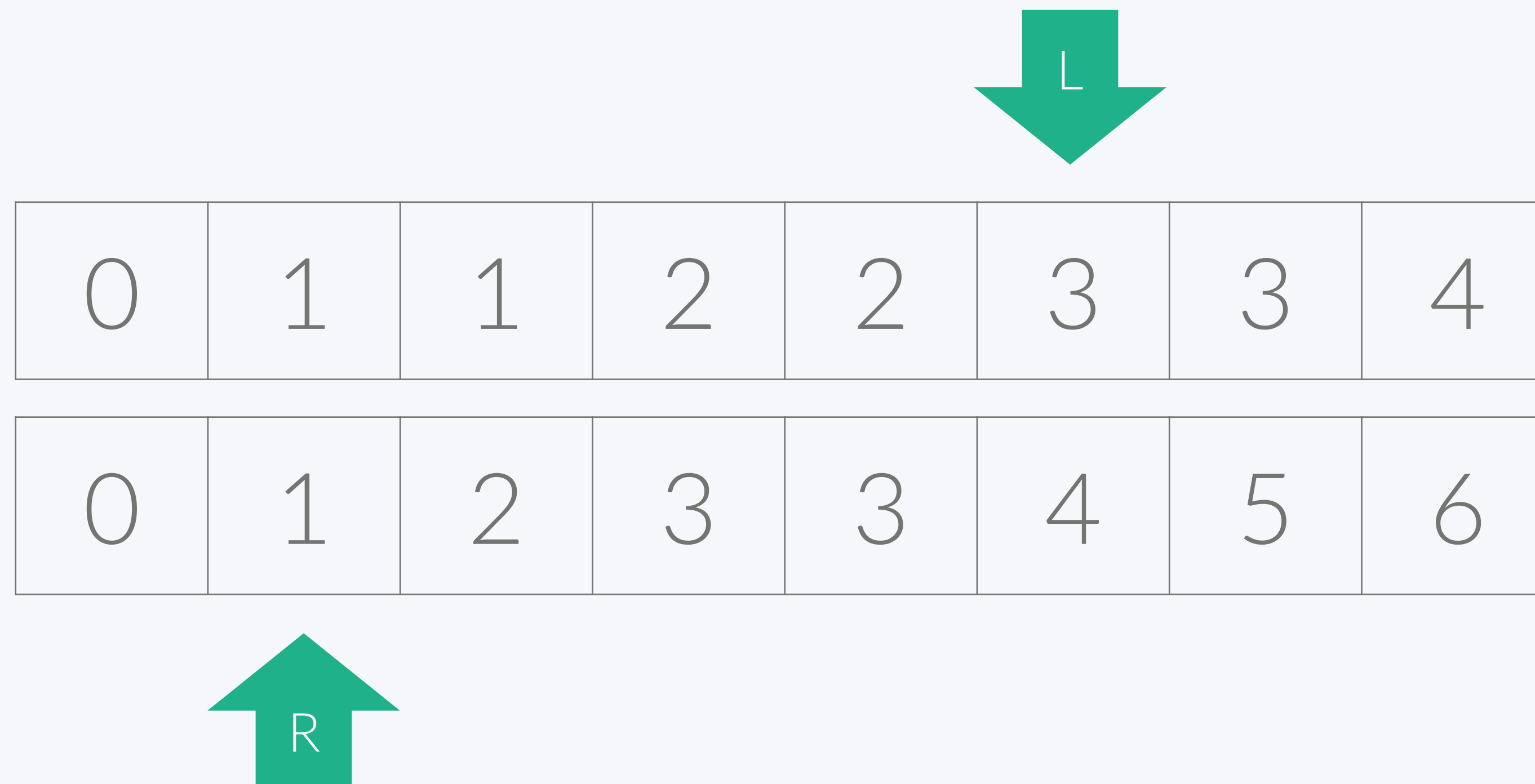


# 부분집합의 합 2

48

<https://www.acmicpc.net/problem/1208>

- $3 + 1 = 4$ 이고, 찾으려고 하는 수 4 이다.
- 위에 3이 2개, 아래에 1이 1개 있기 때문에, 4는 총  $2 \times 1 = 2$ 개이다.
- 이제, L과 R을 이동시킨다.





# 부분집합의 합 2

<https://www.acmicpc.net/problem/1208>



- $4 + 0 = 4$ 이고, 찾으려고 하는 수 4 이다.
- 위에 4가 1개, 아래에 0이 1개 있기 때문에, 4는 총  $1 \times 1 = 1$ 개이다.
- 이제, L과 R을 이동시킨다.

$\Rightarrow$

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 2 | 2 | 3 | 3 | 4 |
| 0 | 1 | 2 | 3 | 3 | 4 | 5 | 6 |



# 부분집합의 합 2

50

<https://www.acmicpc.net/problem/1208>

- C: <https://gist.github.com/Baekjoon/6a01e7aa351497f3d78c>
- C++: <https://gist.github.com/Baekjoon/9a29c36ef0fcc80cbaf5>
- Java: <https://gist.github.com/Baekjoon/30cdb032514859caa106>

# 합이 0인 네 정수

<https://www.acmicpc.net/problem/7453>

- 총 가능한 경우의 수:  $N^4$  가지
- 절반으로 나눠서
- $A[a] + B[b] = N^2$  가지
- $C[c] + D[d] = N^2$  가지
- 계산해볼 수 있다.

51

크기:  $N$

$N^4$

배열  $A, B, C, D$

$$A[a] + B[b] + C[c] + D[d] = 0$$

$$A[a] + B[b] =$$

$$-(C[c] + D[d])$$

# 합이 0인 네 정수

52

<https://www.acmicpc.net/problem/7453>

- 총 가능한 경우의 수:  $N^4$  가지
- 절반으로 나눠서
- $A[a] + B[b] = N^2$  가지
- $C[c] + D[d] = N^2$  가지
- 계산해볼 수 있다.

# 합이 0인 네 정수

53

<https://www.acmicpc.net/problem/7453>

- <https://gist.github.com/Baekjoon/03ecf9044ed5b8c2b154>

# 더 풀어볼 문제

더 풀어보기

- 피자 판매: <https://www.acmicpc.net/problem/2632>
- 두 배열의 합: <https://www.acmicpc.net/problem/2143>