

브루트 포스

최백준 choi@startlink.io

브루트 포스

리모컨

<https://www.acmicpc.net/problem/1107>

- TV 채널을 리모컨을 이용해 바꾸는 문제
- 버튼: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -
- 일부 숫자 버튼이 고장났다.
- 현재 보고 있는 채널: 100
- 이동하려고 하는 채널: N
- 이 때, 리모컨 버튼을 누르는 횟수를 최소로 하는 문제

리모컨

<https://www.acmicpc.net/problem/1107>

- 예를 들어, 5457에 이동하려면 5, 4, 5, 7을 눌러 4번만에 이동할 수 있다.
- 만약, 숫자 버튼 7이 고장났다면, 5, 4, 5, 6, + 또는 5, 4, 5, 8, -를 눌러 5번만에 이동할 수 있다.
- 6, 7, 8이 고장난 경우에는 5, 4, 5, 5, +, + 또는 5, 4, 5, 9, -, -를 눌러 6번만에 이동할 수 있다.
- 5, 6, 7, 8, 9가 고장난 경우에는 ...

리모컨

<https://www.acmicpc.net/problem/1107>

- 예를 들어, 5457에 이동하려면 5, 4, 5, 7을 눌러 4번만에 이동할 수 있다.
- 만약, 숫자 버튼 7이 고장났다면, 5, 4, 5, 6, + 또는 5, 4, 5, 8, -를 눌러 5번만에 이동할 수 있다.
- 6, 7, 8이 고장난 경우에는 5, 4, 5, 5, +, + 또는 5, 4, 5, 9, -, -를 눌러 6번만에 이동할 수 있다.
- 5, 6, 7, 8, 9가 고장난 경우에는 ... 손으로 구하는건 너무 시간이 오래 걸린다.

리모컨

<https://www.acmicpc.net/problem/1107>

- 6, 7, 8이 고장났고, 5457에 이동하고 싶은 경우를 생각해보자.
- 5, 4, 3, 5, +, +, +, 5, 4, -, -, 5, 4, 5, 5, +, + 는 절대 정답이 될 수 없다.

리모컨

<https://www.acmicpc.net/problem/1107>

- 6, 7, 8이 고장났고, 5457에 이동하고 싶은 경우를 생각해보자.
- 5, 4, 3, 5, +, +, +, 5, 4, -, -, 5, 4, 5, 5, +, + 는 절대 정답이 될 수 없다.
- +나 -를 누르다가 숫자 버튼을 누르는 순간 그 전까지 입력한 것이 모두 의미 없어진다.
- 이 문제는 버튼을 누르는 횟수의 최소값을 구하는 문제이다.
- 의미없는 것이 있는 방법은 절대 최소가 될 수 없다.

리모컨

<https://www.acmicpc.net/problem/1107>

- 6, 7, 8이 고장났고, 5457에 이동하고 싶은 경우를 생각해보자.
- 5, 4, 5, 5, -, -, +, +, +, + 는 절대 정답이 될 수 없다.

리모컨

<https://www.acmicpc.net/problem/1107>

- 6, 7, 8이 고장났고, 5457에 이동하고 싶은 경우를 생각해보자.
 - 5, 4, 5, 5, -, -, +, +, +, + 는 절대 정답이 될 수 없다.
 - 이동하는 채널을 적어보면 아래와 같다.
- 5: 5
 - 4: 54
 - 545: 545
 - 5455: 5455
 - 5455-: 5454
 - 5455--: 5453
 - 5455--+: 5454
 - 5455--++: 5455
 - 5455--+++: 5456
 - 5455--++++: 5457

리모컨

10

<https://www.acmicpc.net/problem/1107>

- 6, 7, 8이 고장났고, 5457에 이동하고 싶은 경우를 생각해보자.
 - 5, 4, 5, 5, -, -, +, +, +, + 는 절대 정답이 될 수 없다.
 - 이동하는 채널을 적어보면 아래와 같다.
- 5: 5
 - 4: 54
 - 545: 545
 - 5455: 5455
 - 5455-: 5454
 - 5455--: 5453
 - 5455--+: 5454
 - 5455--++: 5455
 - 5455--+++: 5456
 - 5455--++++: 5457

리모컨

<https://www.acmicpc.net/problem/1107>

- 6, 7, 8이 고장났고, 5457에 이동하고 싶은 경우를 생각해보자.
 - 5, 4, 5, 5, -, -, +, +, +, + 는 절대 정답이 될 수 없다.
 - 이동하는 채널을 적어보면 아래와 같다.
 - 이 문제는 버튼을 누르는 횟수의 최소값을 구하는 문제이다.
 - 중복이 있는 경우는 절대 최소가 될 수 없다.
- 5: 5
 - 4: 54
 - 545: 545
 - 5455: 5455
 - 5455-: 5454
 - 5455--: 5453
 - 5455--+: 5454
 - 5455--++: 5455
 - 5455--+++: 5456
 - 5455--++++: 5457

리모컨

<https://www.acmicpc.net/problem/1107>

- 따라서, 숫자 버튼을 누르고, 그 다음 +나 -중 하나만 연속해서 눌러야 한다.
- 이동하려고 하는 채널 N ($0 \leq N \leq 500,000$)
- 숫자 버튼을 눌러서 이동하는 채널 C 도 $0 \leq C \leq 500,000$ 이면 된다.

리모컨

<https://www.acmicpc.net/problem/1107>

- 따라서, 숫자 버튼을 누르고, 그 다음 +나 -중 하나만 연속해서 눌러야 한다.
- 이동하려고 하는 채널 N ($0 \leq N \leq 500,000$)
- 숫자 버튼을 눌러서 이동하는 채널 C 도 $0 \leq C \leq 500,000$ 이면 된다.
- 아니다

리모컨

<https://www.acmicpc.net/problem/1107>

- 따라서, 숫자 버튼을 누르고, 그 다음 +나 -중 하나만 연속해서 눌러야 한다.
- 이동하려고 하는 채널 N ($0 \leq N \leq 500,000$)
- 숫자 버튼을 눌러서 이동하는 채널 C 도 $0 \leq C \leq 500,000$ 이면 된다.
- 아니다
- 500,000에 이동해야 하고, 1과 5만 누를 수 있는 경우
- 155,555 -> 500,000 보다 (+ 344,445번)
- 511,111 -> 500,000이 더 좋다. (- 11,111번)

리모컨

<https://www.acmicpc.net/problem/1107>

- 따라서, 숫자 버튼을 누르고, 그 다음 +나 -중 하나만 연속해서 눌러야 한다.
- 이동하려고 하는 채널 N ($0 \leq N \leq 500,000$)
- 숫자 버튼을 눌러서 이동하는 채널 C 도 $0 \leq C \leq 1,000,000$ 이면 된다.

리모컨

<https://www.acmicpc.net/problem/1107>

- 가장 처음에 보고 있는 채널은 100이다.
- 따라서, 정답의 최대값은 $500,000 - 100$ 을 넘지않지만, 계산을 편하게 하기 위해 0이라고 가정했다.

리모컨

<https://www.acmicpc.net/problem/1107>

- 숫자 버튼을 이용해 채널 C로 이동한 다음 ($0 \leq C \leq 1,000,000$)
- 거기서 +나 -버튼을 몇 번 눌러야하는지 계산을 해본다
- +나 -를 누르는 횟수 계산은 뱌셈으로 구할 수 있다

리모컨

<https://www.acmicpc.net/problem/1107>

1. 이동할 채널 C를 정한다
2. C에 포함되어있는 숫자 중에 고장난 버튼이 있는지 확인한다
3. 고장난 버튼이 포함되어 있지 않다면 $|C - N|$ 을 계산해 +나 - 버튼을 몇 번 눌러야 하는지를 계산한다

리모컨

<https://www.acmicpc.net/problem/1107>

1. 이동할 채널 C 를 정한다
2. C 에 포함되어있는 숫자 중에 고장난 버튼이 있는지 확인한다
3. 고장난 버튼이 포함되어 있지 않다면 $|C - N|$ 을 계산해 +나 - 버튼을 몇 번 눌러야 하는지를 계산한다

```
for (int i=0; i<=10000000; i++) {  
    int c = i;  
}
```

리모컨

20

<https://www.acmicpc.net/problem/1107>

1. 이동할 채널 C를 정한다
2. C에 포함되어있는 숫자 중에 고장난 버튼이 있는지 확인한다
 - 수를 문자열로 바꾼 다음, 한 글자씩 검사하는 방법
 - 수를 10으로 계속해서 나누면서 하나씩 검사하는 방법
3. 고장난 버튼이 포함되어 있지 않다면 $|C - N|$ 을 계산해 +나 - 버튼을 몇 번 눌러야 하는지를 계산한다

리모컨

<https://www.acmicpc.net/problem/1107>

1. 이동할 채널 C를 정한다
2. C에 포함되어있는 숫자 중에 고장난 버튼이 있는지 확인한다
3. 고장난 버튼이 포함되어 있지 않다면 $|C - N|$ 을 계산해 +나 - 버튼을 몇 번 눌러야 하는지를 계산한다

```
bool broken[10]; // 버튼이 고장나 있으면 true, 아니면 false
bool possible(int c) {
    while (c > 0) {
        if (broken[c % 10]) return false;
        c /= 10;
    }
    return true;
}
```

리모컨

<https://www.acmicpc.net/problem/1107>

- C에 포함되어있는 숫자 중에 고장난 버튼이 있는지 확인한다
- possible(0) 은 항상 true를 리턴한다.

```
bool broken[10]; // 버튼이 고장나 있으면 true, 아니면 false
bool possible(int c) {
    while (c > 0) {
        if (broken[c % 10]) return false;
        c /= 10;
    }
    return true;
}
```

리모컨

<https://www.acmicpc.net/problem/1107>

- 0인 경우를 처리하는 코드

```
if (c == 0) {  
    if (broken[0]) {  
        return false;  
    } else {  
        return true;  
    }  
}
```

리모컨

<https://www.acmicpc.net/problem/1107>

- possible을 불가능하면 0, 가능하면 버튼을 눌러야 하는 횟수를 리턴하게 변경

```
int possible(int c) {  
    if (c == 0) {  
        return broken[0] ? 0 : 1;  
    }  
    int len = 0;  
    while (c > 0) {  
        if (broken[c % 10]) return 0;  
        len += 1;  
        c /= 10;  
    }  
    return len;  
}
```


리모컨

<https://www.acmicpc.net/problem/1107>

- 가장 처음에 보고 있는 채널은 100이기 때문에
- 초기값을 100에서 숫자 버튼을 누르지 않고 이동하는 횟수로 지정

리모컨

26

<https://www.acmicpc.net/problem/1107>

- 소스: <http://codeplus.codes/b70862348c144af7bd2d99929e665351>

건너 뛰며 해보기

카잉 달력

<https://www.acmicpc.net/problem/6064>

- M과 N보다 작거나 같은 두 자연수 x, y 를 이용해서 년도를 $\langle x:y \rangle$ 로 표현한다
- 첫 번째 해는 $\langle 1:1 \rangle$, 두 번째 해는 $\langle 2:2 \rangle$ 이다
- $\langle x:y \rangle$ 의 다음 해는 $\langle x':y' \rangle$ 이다.
 - $x < M$ 이면 $x' = x + 1$, 아니면 $x' = 1$
 - $y < N$ 이면 $y' = y + 1$, 아니면 $y' = 1$
- M, N, x, y가 주어졌을 때, $\langle x:y \rangle$ 이 몇 번째 해인지 구하는 문제

카잉 달력

<https://www.acmicpc.net/problem/6064>

- $1 \leq M, N \leq 40,000$
- 전체 경우의 수는 $MN = 1,600,000,000$ 가지라서 너무 많다.

카잉 달력

30

<https://www.acmicpc.net/problem/6064>

- $M = 5$
- $N = 7$
- 1: $\langle 1, 1 \rangle$
- 2: $\langle 2, 2 \rangle$
- 3: $\langle 3, 3 \rangle$
- 4: $\langle 4, 4 \rangle$
- 5: $\langle 5, 5 \rangle$
- 6: $\langle 1, 6 \rangle$
- 7: $\langle 2, 7 \rangle$
- 8: $\langle 3, 1 \rangle$
- 9: $\langle 4, 2 \rangle$
- 10: $\langle 5, 3 \rangle$
- 11: $\langle 1, 4 \rangle$
- 12: $\langle 2, 5 \rangle$
- 13: $\langle 3, 6 \rangle$
- 14: $\langle 4, 7 \rangle$
- 15: $\langle 5, 1 \rangle$
- 16: $\langle 1, 2 \rangle$
- 17: $\langle 2, 3 \rangle$
- 18: $\langle 3, 4 \rangle$
- 19: $\langle 4, 5 \rangle$
- 20: $\langle 5, 6 \rangle$
- 21: $\langle 1, 7 \rangle$
- 22: $\langle 2, 1 \rangle$
- 23: $\langle 3, 2 \rangle$
- 24: $\langle 4, 3 \rangle$
- 25: $\langle 5, 4 \rangle$
- 26: $\langle 1, 5 \rangle$
- 27: $\langle 2, 6 \rangle$
- 28: $\langle 3, 7 \rangle$
- 29: $\langle 4, 1 \rangle$
- 30: $\langle 5, 2 \rangle$
- 31: $\langle 1, 3 \rangle$
- 32: $\langle 2, 4 \rangle$
- 33: $\langle 3, 5 \rangle$
- 34: $\langle 4, 6 \rangle$
- 35: $\langle 5, 7 \rangle$

카잉 달력

<https://www.acmicpc.net/problem/6064>

- $M = 5$
 - $N = 7$
- | | | | |
|------------|-------------|-------------|-------------|
| • 0: <0,0> | • 10: <0,3> | • 20: <0,6> | • 30: <0,2> |
| • 1: <1,1> | • 11: <1,4> | • 21: <1,0> | • 31: <1,3> |
| • 2: <2,2> | • 12: <2,5> | • 22: <2,1> | • 32: <2,4> |
| • 3: <3,3> | • 13: <3,6> | • 23: <3,2> | • 33: <3,5> |
| • 4: <4,4> | • 14: <4,0> | • 24: <4,3> | • 34: <4,6> |
| • 5: <0,5> | • 15: <0,1> | • 25: <0,4> | |
| • 6: <1,6> | • 16: <1,2> | • 26: <1,5> | |
| • 7: <2,0> | • 17: <2,3> | • 27: <2,6> | |
| • 8: <3,1> | • 18: <3,4> | • 28: <3,0> | |
| • 9: <4,2> | • 19: <4,5> | • 29: <4,1> | |

카잉 달력

32

<https://www.acmicpc.net/problem/6064>

- $M = 5$
 - $N = 7$
 - $x = 3$
 - $y = 2$
- | | | | |
|------------|-------------|-------------|-------------|
| • 0: <0,0> | • 10: <0,3> | • 20: <0,6> | • 30: <0,2> |
| • 1: <1,1> | • 11: <1,4> | • 21: <1,0> | • 31: <1,3> |
| • 2: <2,2> | • 12: <2,5> | • 22: <2,1> | • 32: <2,4> |
| • 3: <3,3> | • 13: <3,6> | • 23: <3,2> | • 33: <3,5> |
| • 4: <4,4> | • 14: <4,0> | • 24: <4,3> | • 34: <4,6> |
| • 5: <0,5> | • 15: <0,1> | • 25: <0,4> | |
| • 6: <1,6> | • 16: <1,2> | • 26: <1,5> | |
| • 7: <2,0> | • 17: <2,3> | • 27: <2,6> | |
| • 8: <3,1> | • 18: <3,4> | • 28: <3,0> | |
| • 9: <4,2> | • 19: <4,5> | • 29: <4,1> | |

카잉 달력

<https://www.acmicpc.net/problem/6064>

- x 를 이용해 모든 해를 고려하지 않고, $i \times M + x$ ($i \geq 0$)의 형태만 조사하면 된다

카잉 달력

<https://www.acmicpc.net/problem/6064>

- 소스: <http://codeplus.codes/52b8d5a503604ba8a4e47f47ca45ebcc>

수 이어 쓰기 1

<https://www.acmicpc.net/problem/1748>

- 1부터 N까지 수를 이어서 쓰면 새로운 하나의 수를 얻게 된다 ($1 \leq N \leq 100,000,000$)
- 1234567891011121314151617181920212223...
- 이 때, 새로운 수는 몇 자리 수일까?

수 이어 쓰기 1

<https://www.acmicpc.net/problem/1748>

- N 이 너무 크기 때문에, 실제로 수를 만드는 것은 너무 시간이 오래 걸린다

수 이어 쓰기 1

<https://www.acmicpc.net/problem/1748>

- N이 너무 크기 때문에, 실제로 수를 만드는 것은 너무 시간이 오래 걸린다
- 총 N개의 수를 하나의 문자열로 만들어야 한다.
- $O(N) * 10$
- 여기서 10은 수의 최대 자릿수를 의미한다.

수 이어 쓰기 1

<https://www.acmicpc.net/problem/1748>

- 수의 자리수별로 나누어서 문제를 해결할 수 있다.
- $N = 120$
- $1 \sim 9$
- $10 \sim 99$
- $100 \sim 120$

수 이어 쓰기 1

<https://www.acmicpc.net/problem/1748>

- 수의 자리수별로 나누어서 문제를 해결할 수 있다.
- $N = 120$
- $1 \sim 9 \Rightarrow (9-1+1) \times 1$
- $10 \sim 99 \Rightarrow (99-10+1) \times 2$
- $100 \sim 120 \Rightarrow (120-100+1) \times 3$

수 이어 쓰기 1

40

<https://www.acmicpc.net/problem/1748>

- 소스: <http://codeplus.codes/72b18a5d0dc0421e915fa75e260bbdb4>

순열 사용하기

부등호

<https://www.acmicpc.net/problem/2529>

- 부등호 기호 <와 >가 나열된 수열 A가 있다
- 기호의 앞 뒤에 한 자리 숫자를 넣어서 모든 부등호 관계를 만족시키려고 한다
- 이 때, 선택된 수는 모두 달라야 한다
- k개의 부등호 관계를 모두 만족시키는 (k+1)개 자리의 정수 중에서 최대값과 최소값을 구하는 문제

부등호

<https://www.acmicpc.net/problem/2529>

- $k \leq 9$
- $k+1 \leq 10$
- 선택된 숫자는 모두 달라야 하기 때문에, 가능한 수의 개수는 $10!$ 개 이다
- 모두 해보고, 가장 큰 수와 작은 수를 구해본다.

부등호

<https://www.acmicpc.net/problem/2529>

- 입력으로 주어진 부등호가 $<$ $>$ 이고
- 가장 큰 수를 구하는 경우를 생각해보자
- 이 부등호에는 숫자가 총 3개 필요하기 때문에, 가장 큰 숫자 3개인 9, 8, 7로 구성되는 것이 정답임을 알 수 있다
- 반대로 가장 작은 수를 구하는 경우에는 0, 1, 2로 구성되는 것이 정답이 된다

부등호

45

<https://www.acmicpc.net/problem/2529>

- 소스: <http://codeplus.codes/f4311b3b1f30479d9b29a2c5adb2bf10>

단어 수학

<https://www.acmicpc.net/problem/1339>

- 단어 수학은 0부터 9까지의 수를 알파벳 하나로 나타낸 것이다
- N개의 단어를 수로 바꾼 다음에, 합의 최대값을 구하는 문제이다 ($1 \leq N \leq 10$)
- 예를 들어, GCF + ACDEB가 문제라고 한다면
- 만들 수 있는 합의 최대값은 99437이다 ($A = 9, B = 4, C = 8, D = 6, E = 5, F = 3, G = 7$)
- 서로 다른 알파벳의 개수는 10개이다

단어 수학

<https://www.acmicpc.net/problem/1339>

- 서로 다른 알파벳이 10개밖에 안되기 때문에, 각각의 알파벳에 0부터 9까지의 숫자를 적절히 대입하면 된다
- 이 때, 합의 최대값을 구하는 문제이기 때문에, 서로 다른 알파벳이 M개라면, 큰 숫자 M개만 넣어서 조사해보면 된다.

단어 수학

<https://www.acmicpc.net/problem/1339>

- 소스: <http://codeplus.codes/bb4313af22da4764a84073ed3d05ea53>

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- N명을 N/2명씩 두 팀으로 나누려고 한다. ($4 \leq N \leq 20$, N은 짝수)
- 두 팀의 능력치를 구한 다음, 차이의 최소값을 구하는 문제
- $S[i][j] = i$ 번 사람과 j 번 사람이 같은 팀에 속했을 때, 팀에 더해지는 능력치
- 팀의 능력치: 팀에 속한 모든 쌍의 $S[i][j]$ 의 합

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- N명을 $N/2$ 명씩 두 팀으로 나누려고 한다. ($4 \leq N \leq 20$, N은 짝수)
- 이것은, 0을 $N/2$ 개, 1을 $N/2$ 개 넣어서 모든 순열을 구하는 것으로 다 해볼 수 있다.

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- 소스: <http://codeplus.codes/c2439746e5214b78aa0b5fa3a92758c9>

백트래킹

백트래킹

Backtracking

- 재귀 함수를 이용해 브루트 포스를 하다 보면, 더 이상 함수 호출이 의미 없는 경우가 있다.
- 이 때, 이런 경우를 제외하고 브루트 포스를 진행하면 백트래킹이라고 한다.

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- N명을 N/2명씩 두 팀으로 나누려고 한다. ($4 \leq N \leq 20$, N은 짝수)
- 두 팀의 능력치를 구한 다음, 차이의 최소값을 구하는 문제
- $S[i][j] = i$ 번 사람과 j 번 사람이 같은 팀에 속했을 때, 팀에 더해지는 능력치
- 팀의 능력치: 팀에 속한 모든 쌍의 $S[i][j]$ 의 합

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- `go(index, first, second)`
 - `index`번째 사람을 어떤 팀에 넣을지 결정해야 함
 - 1번 팀과 2번 팀에 속한 사람이 각각 `first`, `second`에 들어 있음
- 정답을 찾은 경우
 - `index == n`
- 다음 경우
 - 1번 팀: `go(index, first, second)`
 - 2번 팀: `go(index, first, second)`
 - 두 경우 모두 호출 전에 `first` 또는 `second`에 `index`를 넣고, 호출 후에 빼는 과정이 필요

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- 소스: <http://codeplus.codes/e8920e72271043a3a1b34299486b8db1>

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- go(index, first, second)
 - index번째 사람을 어떤 팀에 넣을지 결정해야 함
 - 1번 팀과 2번 팀에 속한 사람이 각각 first, second에 들어 있음
- 정답을 찾은 경우
 - index == n
- 불가능한 경우
 - first의 크기 > n/2
 - second의 크기 > n/2
- 다음 경우
 - 1번 팀: go(index, first, second)
 - 2번 팀: go(index, first, second)
 - 두 경우 모두 호출 전에 first 또는 second에 index를 넣고, 호출 후에 빼는 과정이 필요

스타트와 링크

58

<https://www.acmicpc.net/problem/14889>

- 소스: <http://codeplus.codes/9a635193536e4ad3bcb96930ded78405>

부등호

<https://www.acmicpc.net/problem/2529>

- 부등호 기호 <와 >가 나열된 수열 A가 있다
- 기호의 앞 뒤에 한 자리 숫자를 넣어서 모든 부등호 관계를 만족시키려고 한다
- 이 때, 선택된 수는 모두 달라야 한다
- k개의 부등호 관계를 모두 만족시키는 (k+1)개 자리의 정수 중에서 최대값과 최소값을 구하는 문제

부등호

<https://www.acmicpc.net/problem/2529>

- 재귀함수를 이용해서 만들어 볼 수 있다
- 소스: <http://codeplus.codes/a393d6b85c1c45769eebdef36ee5f07d>

부등호

<https://www.acmicpc.net/problem/2529>

```
void go(int index, string num) {
    if (index == n+1) {
        if (ok(num)) {
            ans.push_back(num);
        }
        return;
    }
    for (int i=0; i<=9; i++) {
        if (check[i]) continue;
        check[i] = true;
        go(index+1, num+to_string(i));
        check[i] = false;
    }
}
```

부등호

<https://www.acmicpc.net/problem/2529>

- 부등호 기호를 만족하는지를 가장 마지막에 모든 수를 결정하고 검사하고 있다
- 첫 번째 수가 9이고, 두 번째 수가 8인 경우에는
- $9 < 8$ 을 만족하지 않기 때문에, 뒤에 무엇이 온다고 해도 절대로 정답이 될 수 없다
- 하지만 앞 페이지의 소스는 계속해서 검사를 진행한다.

부등호

<https://www.acmicpc.net/problem/2529>

- 함수의 호출 중간에 절대로 정답이 될 수 없는 경우를 발견하면
- 그 뒤의 호출을 더 이상 진행하지 않아도 된다

부등호

<https://www.acmicpc.net/problem/2529>

```
void go(int index, string num) {
    if (index == n+1) {
        ans.push_back(num);
        return;
    }
    for (int i=0; i<=9; i++) {
        if (check[i]) continue;
        if (index == 0 || good(num[index-1], i+'0', a[index-1])) {
            check[i] = true;
            go(index+1, num+to_string(i));
            check[i] = false;
        }
    }
}
```


부등호

<https://www.acmicpc.net/problem/2529>

```
bool good(char x, char y, char op) {  
    if (op == '<') {  
        if (x > y) return false;  
    }  
    if (op == '>') {  
        if (x < y) return false;  
    }  
    return true;  
}
```

부등호

<https://www.acmicpc.net/problem/2529>

- 소스: <http://codeplus.codes/f05284f509be446d804f1a5af795c76b>
- C++: 920MS -> 8MS
- Java: 1260MS -> 196MS

맞춰봐

<https://www.acmicpc.net/problem/1248>

- -10부터 10까지 N개의 정수(중복 없음)로 이루어진 수열 A가 있다. ($N \leq 10$)
- $S[i][j] = A[i] + A[i+1] + \dots + A[j]$ 가 0보다 크면 +, 작으면 -, 같으면 0
- S가 주어졌을 때, 가능한 A를 아무거나 찾는 문제

맞춰봐

<https://www.acmicpc.net/problem/1248>

- 21개의 수를 10개의 자리에 넣어야 한다.
- 총 경우의 수: $21^{10} = 16,679,880,978,201$
- 생각보다 경우의 수가 많다.
- 일단 함수를 작성해보자.

맞춰봐

<https://www.acmicpc.net/problem/1248>

```
bool go(int index) {  
    if (index == n) {  
        return ok();  
    }  
    for (int i=-10; i<=10; i++) {  
        ans[index] = i;  
        if (go(index+1)) return true;  
    }  
    return false;  
}
```

맞춰봐

70

<https://www.acmicpc.net/problem/1248>

- 소스: <http://codeplus.codes/df6d0763f4cb4743a1afe4be80df1635>

맞춰봐

<https://www.acmicpc.net/problem/1248>

- `sign[i][i]`에는 i 번째 수의 부호가 들어있다.
- -10까지 10까지 순회하지 않고
- 양수인 경우에는 1~10
- 음수인 경우에는 -10~-1
- 0인 경우에는 0
- 을 넣는 방식으로 개선해볼 수 있다.

맞춰봐

72

<https://www.acmicpc.net/problem/1248>

```
bool go(int index) {
    if (index == n) return ok();
    if (sign[index][index] == 0) {
        ans[index] = 0;
        return go(index+1);
    }
    for (int i=1; i<=10; i++) {
        ans[index] = sign[index][index]*i;
        if (go(index+1)) return true;
    }
    return false;
}
```


맞춰봐

73

<https://www.acmicpc.net/problem/1248>

- 소스: <http://codeplus.codes/b7d33e6f0a8448bfa6612d9527ce5624>

맞춰봐

<https://www.acmicpc.net/problem/1248>

- index번째 수를 결정하면, 0~index번째 수는 변하지 않는다.
- 따라서, 모든 $\text{sign}[k][\text{index}]$ ($0 \leq k < \text{index}$) 를 $\text{go}(\text{index})$ 에서 검사할 수 있다

맞춰봐

75

<https://www.acmicpc.net/problem/1248>

```
bool go(int index) {
    if (index == n) return true;
    if (sign[index][index] == 0) {
        ans[index] = 0;
        return check(index) && go(index+1);
    }
    for (int i=1; i<=10; i++) {
        ans[index] = sign[index][index]*i;
        if (check(index) && go(index+1)) return true;
    }
    return false;
}
```

맞춰봐

<https://www.acmicpc.net/problem/1248>

```
bool check(int index) {
    int sum = 0;
    for (int i=index; i>=0; i--) {
        sum += ans[i];
        if (sign[i][index] == 0) {
            if (sum != 0) return false;
        } else if (sign[i][index] < 0) {
            if (sum >= 0) return false;
        } else if (sign[i][index] > 0) {
            if (sum <= 0) return false;
        }
    }
    return true;
}
```

맞춰봐

<https://www.acmicpc.net/problem/1248>

- 소스: <http://codeplus.codes/d54011facdfa4c339af1bc961555d82f>

N-Queen

78

<https://www.acmicpc.net/problem/9663>

- $N \times N$ 크기의 체스판 위에 Queen을 N 개 놓는 방법의 수를 구하는 문제

N-Queen

79

<https://www.acmicpc.net/problem/9663>

- calc(row): row 행에 퀸을 어디에 놓을지 결정해야 함

N-Queen

<https://www.acmicpc.net/problem/9663>

- calc(row): row 행에 퀸을 어디에 놓을지 결정해야 함

```
void calc(int row) {  
    if (row == n) {  
        ans += 1;  
    }  
    for (int col=0; col<n; col++) {  
        a[row][col] = true;  
        if (check(row, col)) {  
            calc(row+1);  
        }  
        a[row][col] = false;  
    }  
}
```


N-Queen

81

<https://www.acmicpc.net/problem/9663>

- 소스: <http://codeplus.codes/0b023b3089ee4a60a084d60e0b95d627>

N-Queen

<https://www.acmicpc.net/problem/9663>

- `check_col[i] = i번 열에 퀸이 놓여져 있으면 true`

	0	1	2	3	4	5
0	0	1	2	3	4	5
1	0	1	2	3	4	5
2	0	1	2	3	4	5
3	0	1	2	3	4	5
4	0	1	2	3	4	5
5	0	1	2	3	4	5

N-Queen

<https://www.acmicpc.net/problem/9663>

- `check_dig[i] = /` 대각선에 퀸이 있으면

	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	6
2	2	3	4	5	6	7
3	3	4	5	6	7	8
4	4	5	6	7	8	9
5	5	6	7	8	9	10

N-Queen

<https://www.acmicpc.net/problem/9663>

- `check_dig2[i] = \` 대각선에 퀸이 있으면

	0	1	2	3	4	5
0	5	4	3	2	1	0
1	6	5	4	3	2	1
2	7	6	5	4	3	2
3	8	7	6	5	4	3
4	9	8	7	6	5	4
5	10	9	8	7	6	5

N-Queen

<https://www.acmicpc.net/problem/9663>

- Check 부분을 배열을 이용하면 놓을 수 있는지 검사를 $O(1)$ 만에 해결 할 수 있다.
- 소스: <http://codeplus.codes/848c31d56c184696b2dc122595da4a90>

스도쿠

<https://www.acmicpc.net/problem/2580>

- 스도쿠를 푸는 문제

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

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

스도쿠

<https://www.acmicpc.net/problem/2580>

- $go(z)$: z 번째 칸을 채우는 함수
- $(x, y) \rightarrow 9 * x + y$ 번째 칸

0	1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44
45	46	47	48	49	50	51	52	53
54	55	56	57	58	59	60	61	62
63	64	65	66	67	68	69	70	71
72	73	74	75	76	77	78	79	80

스도쿠

<https://www.acmicpc.net/problem/2580>

- $c3[i][j] = i$ 번 작은 정사각형에 숫자 j 가 있으면 true
- (x, y) 는 $(x/3)*3+(y/3)$ 번째 칸

0	0	0	1	1	1	2	2	2
0	0	0	1	1	1	2	2	2
0	0	0	1	1	1	2	2	2
3	3	3	4	4	4	5	5	5
3	3	3	4	4	4	5	5	5
3	3	3	4	4	4	5	5	5
6	6	6	7	7	7	8	8	8
6	6	6	7	7	7	8	8	8
6	6	6	7	7	7	8	8	8

<https://www.acmicpc.net/problem/2580>

```
for (int i=0; i<n; i++) {
    for (int j=0; j<n; j++) {
        cin >> a[i][j];
        if (a[i][j] != 0) {
            c[i][a[i][j]] = true;
            c2[j][a[i][j]] = true;
            c3[square(i,j)][a[i][j]] = true;
        }
    }
}

go(0);
```

스도쿠

<https://www.acmicpc.net/problem/2580>

```
void go(int z) {
    if (z == 81) {
        // check
        exit(0);
    }
    int x = z/n, y = z%n;
    if (a[x][y] != 0) {
        go(z+1);
    } else {
        // next
    }
}
```

<https://www.acmicpc.net/problem/2580>

```
// check
for (int i=0; i<n; i++) {
    for (int j=0; j<n; j++) {
        cout << a[i][j] << ' ';
    }
    cout << '\n';
}
exit(0);
```

<https://www.acmicpc.net/problem/2580>

```
// next
```

```
for (int i=1; i<=9; i++) {  
    if (c[x][i] == 0 && c2[y][i] == 0 && c3[square(x,y)][i]==0) {  
        c[x][i] = c2[y][i] = c3[square(x,y)][i] = true;  
        a[x][y] = i;  
        go(z+1);  
        a[x][y] = 0;  
        c[x][i] = c2[y][i] = c3[square(x,y)][i] = false;  
    }  
}
```

스도쿠

<https://www.acmicpc.net/problem/2580>

- 소스: <http://codeplus.codes/94881686f4e4468394c59e0b95f85717>

알파벳

<https://www.acmicpc.net/problem/1987>

- 세로 R칸, 가로 C칸으로 된 표 모양의 보드가 있다
- 보드의 각 칸에는 대문자 알파벳이 하나씩 적혀 있고, 좌측 상단 칸 (1행 1열) 에는 말이 놓여 있다
- 말은 상하좌우로 인접한 네 칸 중의 한 칸으로 이동할 수 있다
- 같은 알파벳이 적힌 칸을 두 번 지날 수 없다
- 좌측 상단에서 시작해서, 말이 최대한 몇 칸을 지날 수 있는지를 구하는 문제

알파벳

<https://www.acmicpc.net/problem/1987>

- go(board, check, x, y, cnt)
 - board: 보드
 - check: 방문한 알파벳
 - x, y: 현재 위치
 - cnt: 방문한 칸의 수

알파벳

<https://www.acmicpc.net/problem/1987>

- `go(board, check, x, y, cnt)`
 - board: 보드
 - check: 방문한 알파벳
 - x, y: 현재 위치
 - cnt: 방문한 칸의 수
- 새로운 칸 `nx, ny`로 이동할 수 있는 경우
 - `go(board, check, nx, ny, cnt+1)`
 - 이 때, check는 변경해 줘야함

알파벳

<https://www.acmicpc.net/problem/1987>

```
void go(vector<string> &board, vector<bool> &check, int x, int y, int
cnt) {
    if (cnt > ans) ans = cnt;
    for (int k=0; k<4; k++) {
        int nx = x+dx[k];
        int ny = y+dy[k];
        if (nx >= 0 && nx < board.size() && ny >= 0 && ny <
board[0].size()) {
            if (check[board[nx][ny]-'A'] == false) {
                check[board[nx][ny]-'A'] = true;
                go(board, check, nx, ny, cnt+1);
                check[board[nx][ny]-'A'] = false;
            }
        }
    }
}
```

알파벳

100

<https://www.acmicpc.net/problem/1987>

- `go(board, check, x, y)`
- `board`: 보드
- `check`: 방문한 알파벳
- `x, y`: 현재 위치
- 리턴 값: 방문할 수 있는 칸의 최대 개수
- 의미: (x, y) 에서 이동을 시작하고, 방문한 알파벳이 `check`일 때, 방문할 수 있는 칸의 최대 개수

알파벳

<https://www.acmicpc.net/problem/1987>

```
int go(vector<string> &board, vector<bool> &check, int x, int y) {
    int ans = 0;
    for (int k=0; k<4; k++) {
        int nx = x+dx[k], ny = y+dy[k];
        if (nx >= 0 && nx < board.size() && ny >= 0 && ny <
board[0].size()) {
            if (check[board[nx][ny]-'A'] == false) {
                check[board[nx][ny]-'A'] = true;
                int next = go(board, check, nx, ny);
                if (ans < next) ans = next;
                check[board[nx][ny]-'A'] = false;
            }
        }
    }
    return ans + 1;
}
```

알파벳

102

<https://www.acmicpc.net/problem/1987>

- 소스: <http://codeplus.codes/f292112033e24243bbb323775c1d3bf6>

비트마스크

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- N명을 $N/2$ 명씩 두 팀으로 나누려고 한다. ($4 \leq N \leq 20$, N은 짝수)
- 두 팀의 능력치를 구한 다음, 차이의 최소값을 구하는 문제
- $S[i][j] = i$ 번 사람과 j 번 사람이 같은 팀에 속했을 때, 팀에 더해지는 능력치
- 팀의 능력치: 팀에 속한 모든 쌍의 $S[i][j]$ 의 합

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- 각 사람을 두 팀 중 하나로 나누는 문제이기 때문에, 비트마스크를 이용할 수 있다.
- 비트가 0인 사람은 0번 팀, 1인 사람은 1번 팀이라고 하고
- 전체 경우의 수를 순회할 수 있다.

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- 소스 1: <http://codeplus.codes/aeb963d5ccad4c799781681f36e94885>
- 소스 2: <http://codeplus.codes/e4c245f953e643f39b8157e0a27df20e>

종이 조각

<https://www.acmicpc.net/problem/14391>

- $N \times M$ 크기의 종이를 조각으로 잘라서 합의 최대값을 구하는 문제 ($1 \leq N, M \leq 4$)

4	9	3	7
2	5	9	1
3	8	4	6
9	1	5	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 종이를 조각으로 잘라서 합의 최대값을 구하는 문제

4 9 3			7
2	5	9	1
3	8	4	6
9	1	5	0

종이 조각

109

<https://www.acmicpc.net/problem/14391>

- 어차피 3자리 수는 항상 4자리 수보다 작다
- 정답은 네자리 수 4개 합이 아닐까?

종이 조각

110

<https://www.acmicpc.net/problem/14391>

- 어차피 3자리 수는 항상 4자리 수보다 작다
- 정답은 네자리 수 4개 합이 아닐까?

0	0	0	5
0	0	0	0
0	0	0	0
5	0	0	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 어차피 3자리 수는 항상 4자리 수보다 작다
- 정답은 네자리 수 4개 합이 아닐까?

0	0	0	5
0	0	0	0
0	0	0	0
5	0	0	0

0	0	0	5
0	0	0	0
0	0	0	0
5	0	0	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 어차피 3자리 수는 항상 4자리 수보다 작다
- 정답은 네자리 수 4개 합이 아닐까? 아니다
- 반례가 있다

0	0	0	5
0	0	0	0
0	0	0	0
5	0	0	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- $1 \leq N, M \leq 4$ 이다.
- 각각의 칸은 가로 또는 세로 칸에 속하게 된다

4	9	3	7
2	5	9	1
3	8	4	6
9	1	5	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 각각의 칸에 대해서, 가로(-)인지 세로(|)인지 정하면 된다.

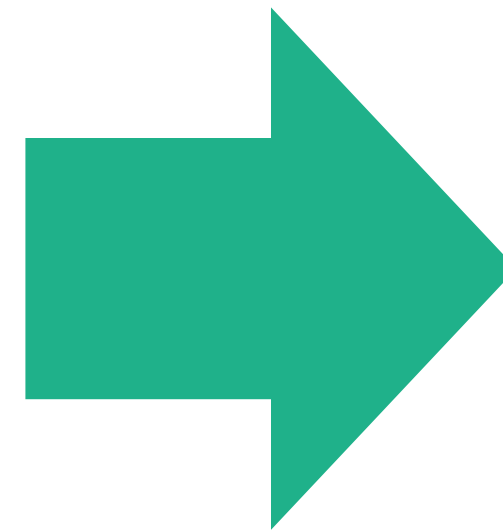
-	-		
-	-	-	
	-		
	-		-

종이 조각

<https://www.acmicpc.net/problem/14391>

- 각각의 칸에 대해서, 가로(-)인지 세로(|)인지 정하면 된다.

-	-		
-	-	-	
	-		
	-		-



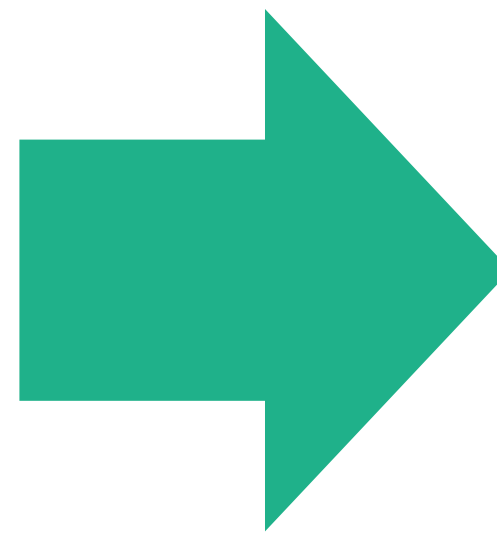
0	0	1	1
0	0	0	1
1	0	1	1
1	0	1	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 각각의 칸에 대해서, 가로(-)인지 세로(|)인지 정하면 된다.

-	-		
-	-	-	
	-		
	-		-



0	0	1	1
0	0	0	1
1	0	1	1
1	0	1	0

0	0	1	1	0	0	0	1	1	0	1	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

종이 조각

117

<https://www.acmicpc.net/problem/14391>

- $2^{(NM)}$ 으로 상태를 만들고 나누어보면 된다

종이 조각

118

<https://www.acmicpc.net/problem/14391>

- 소스: <http://codeplus.codes/0a6e3902bdfd4003b7355a49f2ac6021>

가르침

119

<https://www.acmicpc.net/problem/1062>

- N개의 단어가 주어졌을 때
- K개의 글자로만 이루어진 단어의 개수를 고르는 문제
- 모든 단어는 anta로 시작하고
- 모든 단어는 tica로 끝난다
- $N \leq 50$, 단어의 길이 ≤ 15

가르침

120

<https://www.acmicpc.net/problem/1062>

- N개의 단어가 주어졌을 때
- K개의 글자로만 이루어진 단어의 개수를 고르는 문제
- 모든 단어는 **anta**로 시작하고
- 모든 단어는 **tica**로 끝난다
- $N \leq 50$, 단어의 길이 ≤ 15
- 먼저, a, n, t, i, c는 가르쳐야 한다
- 즉, 26-5개의 글자 중에서 K-5개를 고르는 문제

가르침

121

<https://www.acmicpc.net/problem/1062>

- 경우의 수가 몇 개 되지 않기 때문에, 모든 경우를 다 시뮬레이션 해본다

가르침

122

<https://www.acmicpc.net/problem/1062>

```
int go(int index, int k, vector<string> &words) {
    if (k < 0) return 0;
    if (index == 26) return count(words);
    int ans = 0;
    learn[index] = true;
    int t1 = go(index+1, k-1, words);
    learn[index] = false;
    if (ans < t1) ans = t1;
    if (index != 'a'-'a' && index != 'n'-'a' && index != 't'-'a'
    && index != 'i'-'a' && index != 'c'-'a') {
        t1 = go(index+1, k, words);
        if (ans < t1) ans = t1;
    }
    return ans;
}
```

가르침

123

<https://www.acmicpc.net/problem/1062>

```
int count(vector<string> &words) {  
    int cnt = 0;  
    for (string word : words) {  
        bool ok = true;  
        for (char x : word) {  
            if (!learn[x-'a']) {  
                ok = false;  
                break;  
            }  
        }  
        if (ok) cnt += 1;  
    }  
    return cnt;  
}
```

가르침

124

<https://www.acmicpc.net/problem/1062>

- 소스: <http://codeplus.codes/2ae7c60b801f47c480fa5576c86a1512>

가르침

125

<https://www.acmicpc.net/problem/1062>

- 각각의 단어가 배운 알파벳으로만 이루어져 있는지 검사하는 것이 너무 오랜 시간이 걸린다.
- $O(\text{단어의 개수} \times \text{각 단어의 길이})$
- 이 부분을 $O(\text{단어의 개수})$ 로 줄일 수 있다.

가르침

<https://www.acmicpc.net/problem/1062>

- 실제로 그 단어가 무엇인지가 중요한 것이 아니다.
- 그 단어에 알파벳이 어떤 순서로 이루어져 있는지 중요한 것이 아니다.
- 각 단어에 속해있는 알파벳이 무엇인지만 중요하다.

가르침

127

<https://www.acmicpc.net/problem/1062>

- 각 단어를 비트마스크로 나타낼 수 있다.
- $\text{word}[i]$ = i 번째 단어에 포함되어 있는 알파벳의 비트마스크

가르침

128

<https://www.acmicpc.net/problem/1062>

```
vector<int> words(n);  
for (int i=0; i<n; i++) {  
    string s;  
    cin >> s;  
    for (char x : s) {  
        words[i] |= (1 << (x-'a'));  
    }  
}
```


가르침

<https://www.acmicpc.net/problem/1062>

```
int go(int index, int k, int mask, vector<int> &words) {
    if (k < 0) return 0;
    if (index == 26) return count(mask, words);
    int ans = 0;
    int t1 = go(index+1, k-1, mask | (1 << index), words);
    if (ans < t1) ans = t1;
    if (index != 'a'-'a' && index != 'n'-'a' && index != 't'-'a'
    && index != 'i'-'a' && index != 'c'-'a') {
        t1 = go(index+1, k, mask, words);
        if (ans < t1) ans = t1;
    }
    return ans;
}
```

가르침

130

<https://www.acmicpc.net/problem/1062>

```
int count(int mask, vector<int> &words) {  
    int cnt = 0;  
    for (int word : words) {  
        if ((word & ((1<<26)-1-mask)) == 0) {  
            cnt += 1;  
        }  
    }  
    return cnt;  
}
```

가르침

<https://www.acmicpc.net/problem/1062>

- 소스: <http://codeplus.codes/3ecbd3de00e04cc8ad6fc94b25347009>

구슬 탈출 2

132

<https://www.acmicpc.net/problem/13460>

- 보드의 상태가 주어졌을 때, 최소 몇 번 만에 빨간 구슬을 구멍을 통해 빼낼 수 있는지 구하는 문제
- 만약, 10번 이내에 움직여서 빨간 구슬을 구멍을 통해 빼낼 수 없으면 -1을 출력

구슬 탈출 2

133

<https://www.acmicpc.net/problem/13460>

- 보드의 상태가 주어졌을 때, 최소 몇 번 만에 빨간 구슬을 구멍을 통해 빼낼 수 있는지 구하는 문제
- 만약, **10번 이내**에 움직여서 빨간 구슬을 구멍을 통해 빼낼 수 없으면 -1을 출력

구슬 탈출 2

134

<https://www.acmicpc.net/problem/13460>

- 이동할 수 있는 방향이 4방향
- 최대 10번 이내로 움직여야 한다
- 가능한 이동 방법의 수: $4^{10} = 1,048,576$

구슬 탈출 2

135

<https://www.acmicpc.net/problem/13460>

- 같은 방향으로 연속해서 두 번 이상 이동하는건 의미가 없다
- 가능한 이동 방법의 수: $4 * 3^9 = 78,732$ 가지

구슬 탈출 2

136

<https://www.acmicpc.net/problem/13460>

- 같은 방향으로 연속해서 두 번 이상 이동하는건 의미가 없다
- 한 방향으로 이동한 다음, 반대 방향으로 바로 이동하는 것도 의미가 없다
- 가능한 이동 방법의 수: $4 * 2^9 = 2,048$ 가지

구슬 탈출 2

137

<https://www.acmicpc.net/problem/13460>

- 먼저, 이동 가능한 방법을 비트마스크를 이용해서 4^{10} 가지를 만든 다음
- 앞 페이지에 나온 두 가지 경우를 모두 제외시킨다
- 4^{10} 을 만들기 위해 0부터 2^{20} 까지 수를 모두 만들고
- 4진법으로 변환해서 경우의 수를 모두 만든다

구슬 탈출 2

138

<https://www.acmicpc.net/problem/13460>

- 그 다음, 문제에 나와있는 대로 시뮬레이션 해본다.
- 동시에 두 개의 공을 이동시키는 것은 어렵기 때문에
- 공을 하나씩 움직여서 더 이상 두 공이 움직이지 않을 때까지 이동시켜본다

구슬 탈출 2

139

<https://www.acmicpc.net/problem/13460>

- 소스: <http://codeplus.codes/54b22afc22094b55a550a32770371b25>

2048 (Easy)

<https://www.acmicpc.net/problem/12100>

140

- 2048 게임에서 최대 5번 이동시켜서 얻을 수 있는 가장 큰 블록을 출력하는 문제

2048 (Easy)

<https://www.acmicpc.net/problem/12100>

141

- 이동 횟수가 5번이기 때문에, $4^5 = 1024$ 번 이동을 시켜보면 된다.

2048 (Easy)

142

<https://www.acmicpc.net/problem/12100>

- 소스: <http://codeplus.codes/a95d8abd00a24b6ba53935bdac95d746>

일부 경우만 해보기

일부 경우만 해보기

144

Two Pointers Algorithm

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

수들의 합 2

145

<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

수들의 합 2

146

<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

147

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

- 찾으려고 하는 수: 5
- 합: 1



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



수들의 합 2

148

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

- 찾으려고 하는 수: 5
- 합: 3



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



수들의 합 2

149

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

- 찾으려고 하는 수: 5
- 합: 6

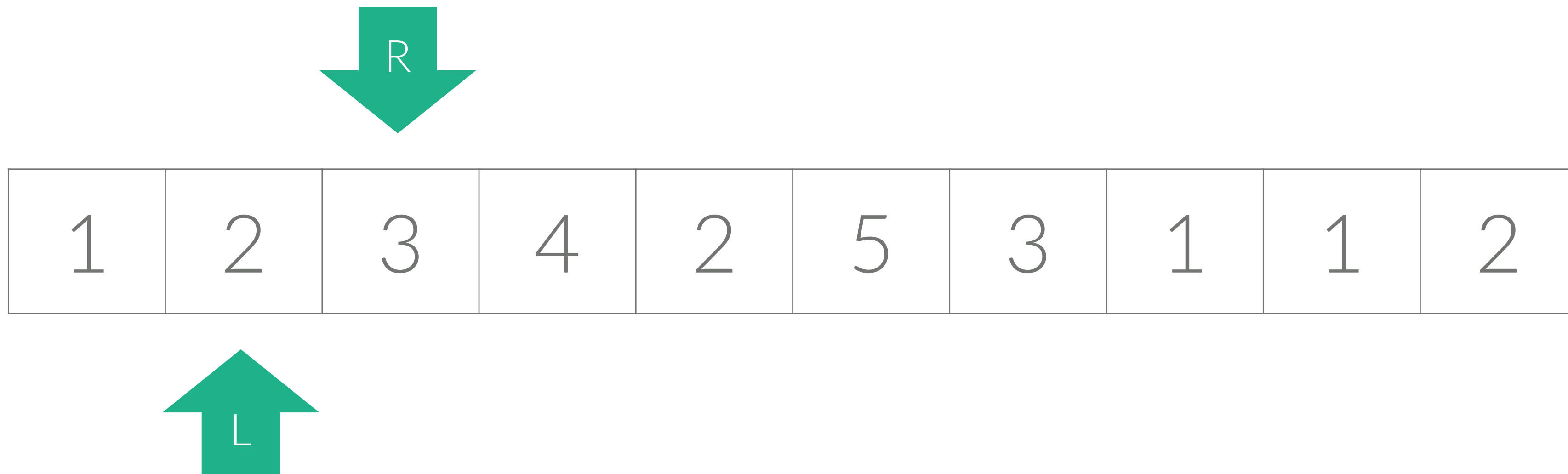


수들의 합 2

150

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

- 찾으려고 하는 수: 5
- 합: 5 (찾았다!)

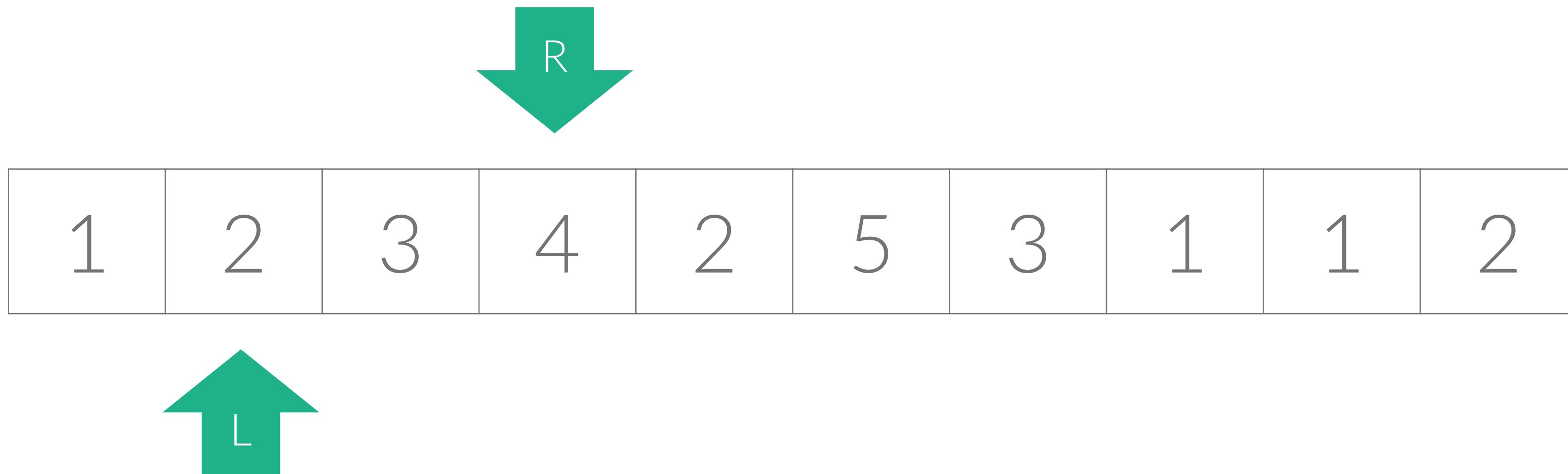


수들의 합 2

151

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

- 찾으려고 하는 수: 5
- 합: 9

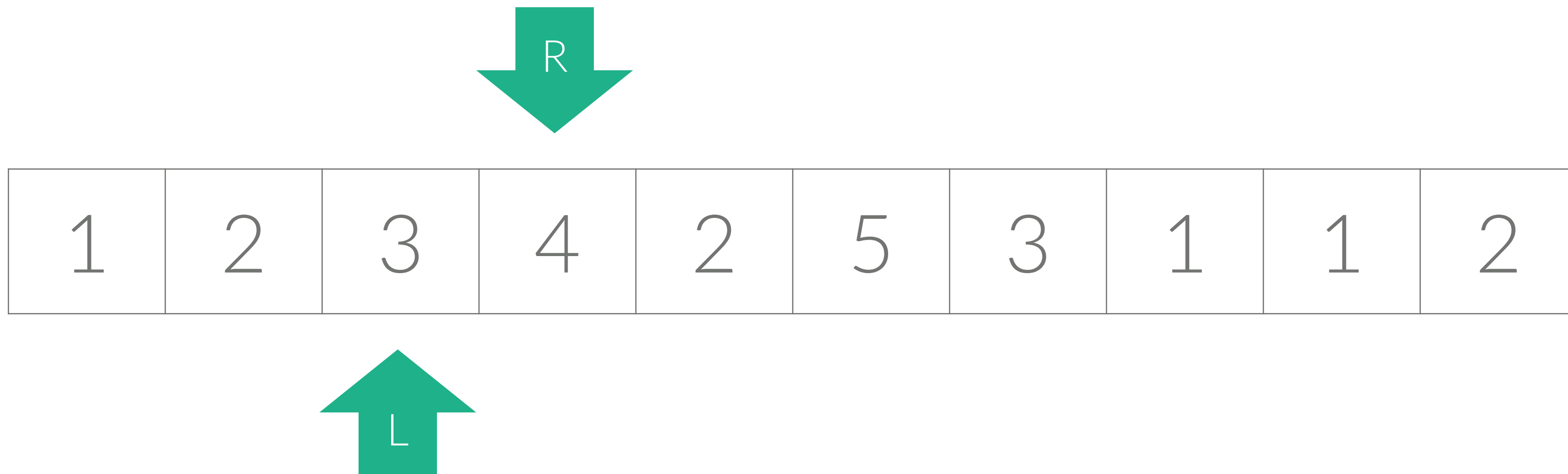


수들의 합 2

152

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

- 찾으려고 하는 수: 5
- 합: 7

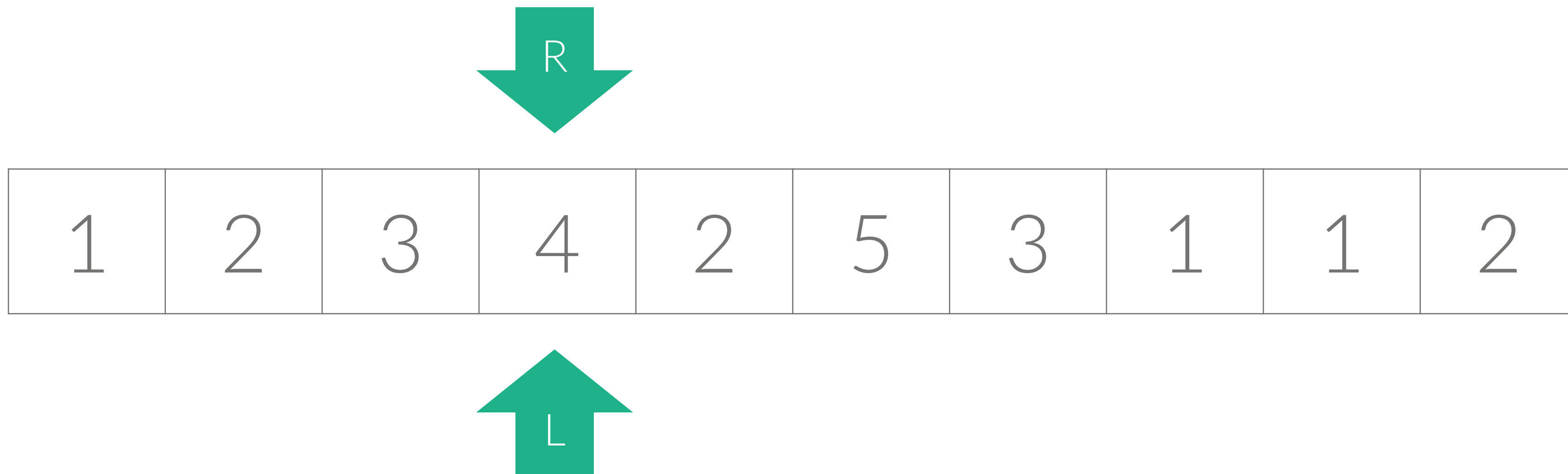


수들의 합 2

153

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

- 찾으려고 하는 수: 5
- 합: 4

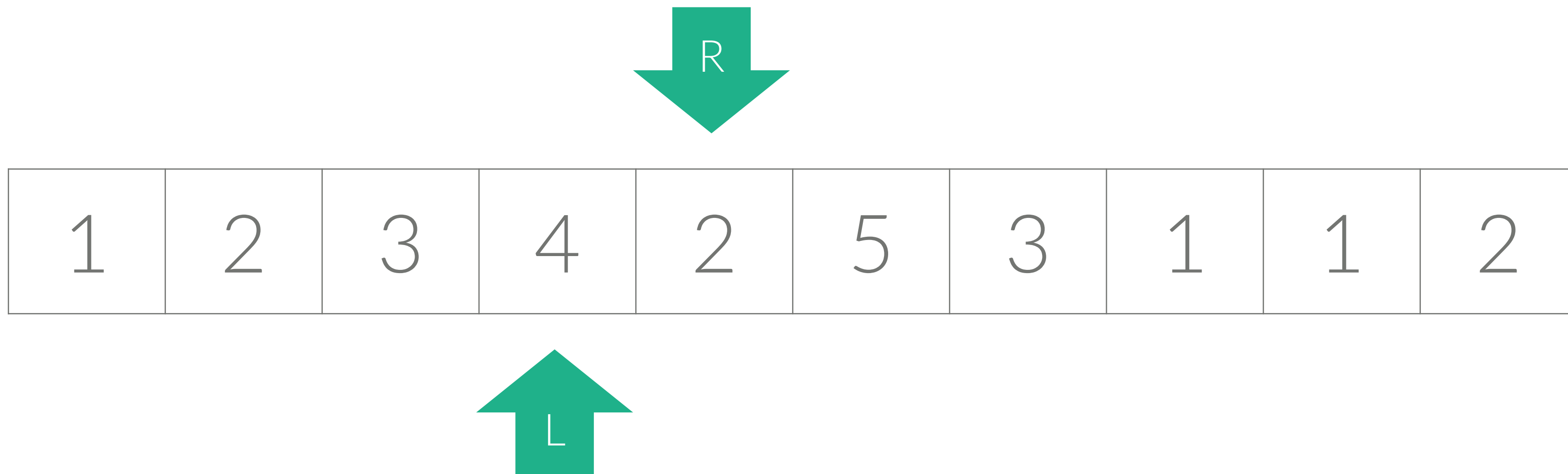


수들의 합 2

154

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

- 찾으려고 하는 수: 5
- 합: 6

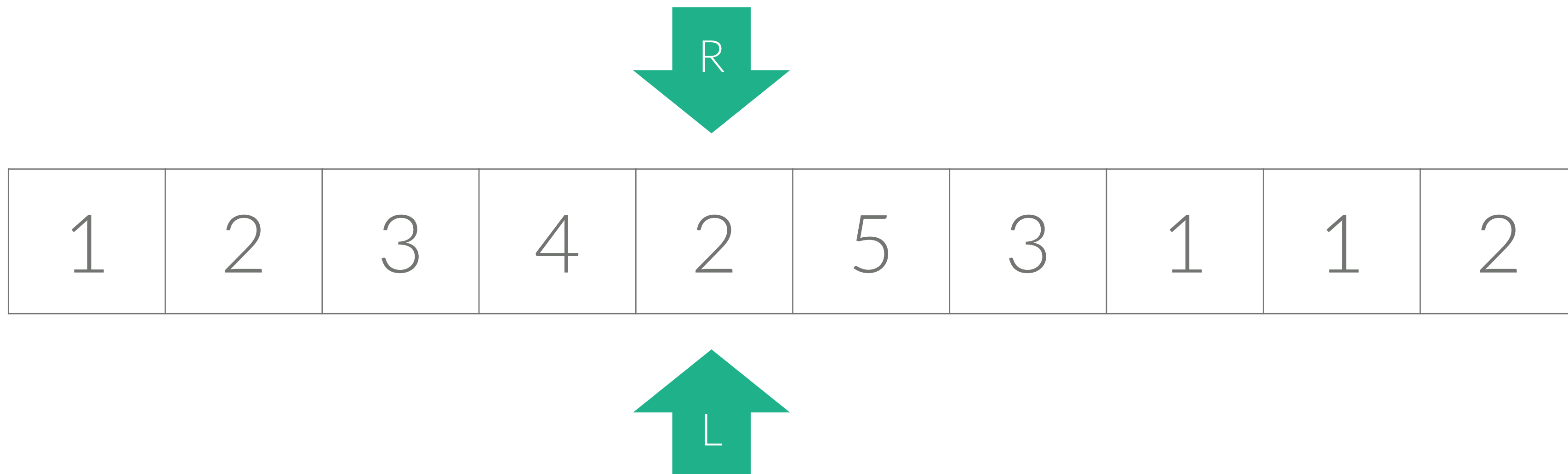


수들의 합 2

155

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

- 찾으려고 하는 수: 5
- 합: 2

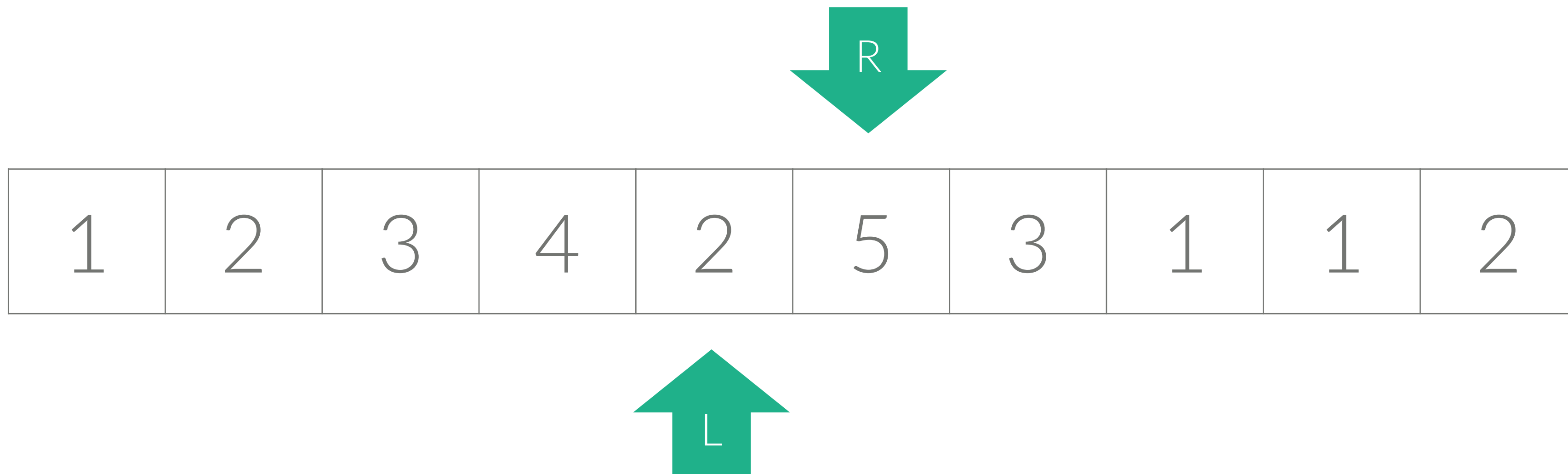


수들의 합 2

156

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

- 찾으려고 하는 수: 5
- 합: 7



수들의 합 2

157

<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
---	---	---	---	---	---	---	---	---	---

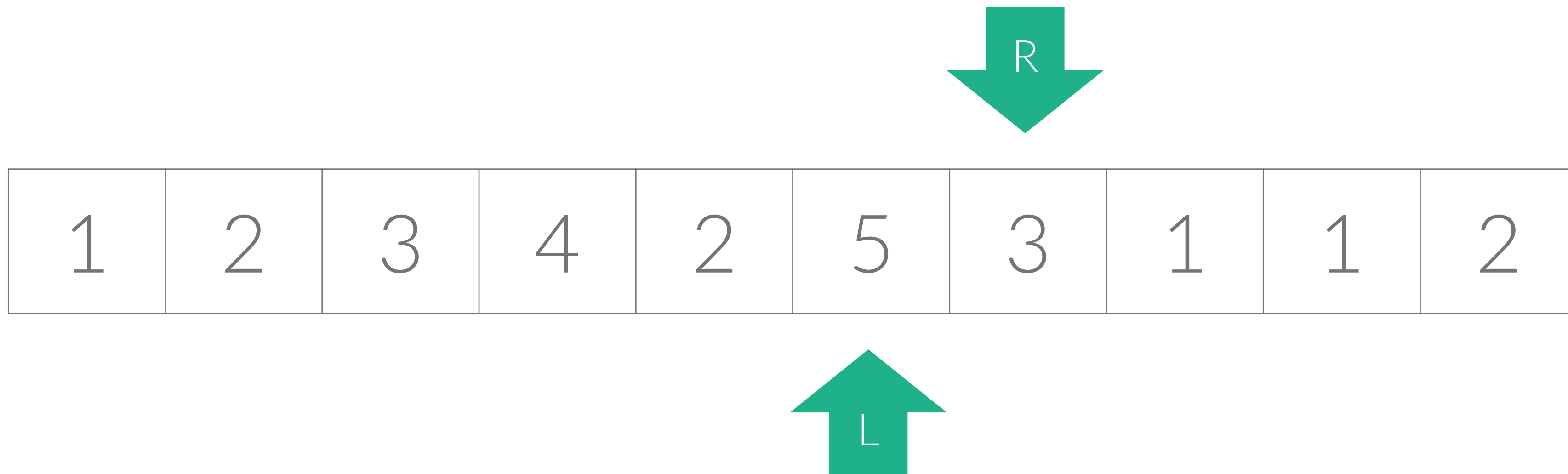


수들의 합 2

158

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

- 찾으려고 하는 수: 5
- 합: 8

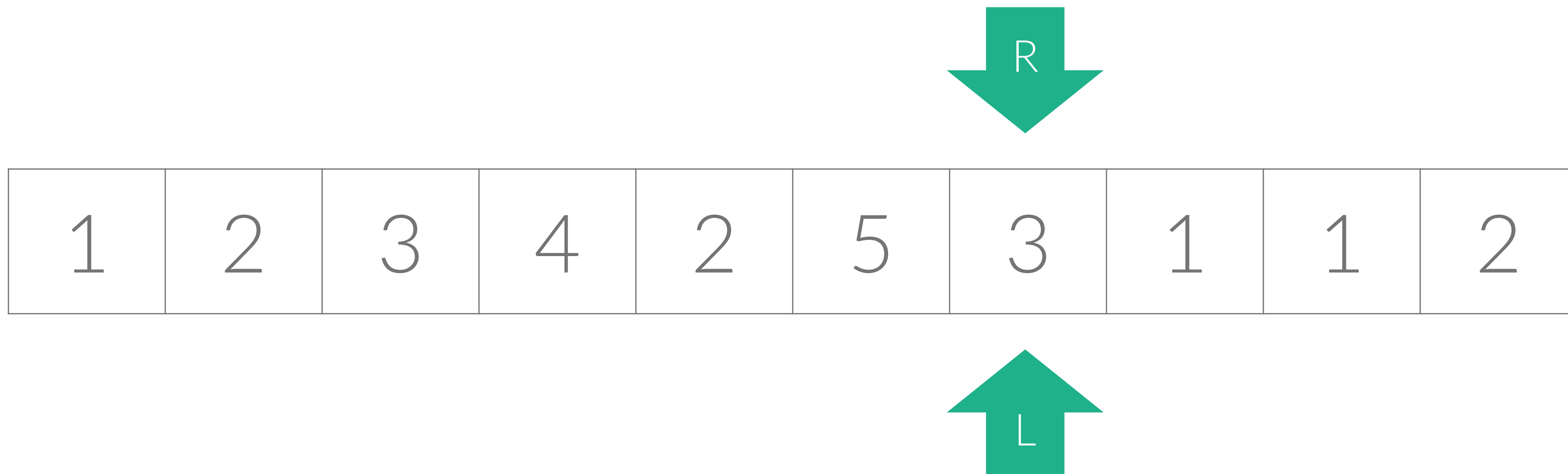


수들의 합 2

159

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

- 찾으려고 하는 수: 5
- 합: 3



수들의 합 2

160

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

- 찾으려고 하는 수: 5
- 합: 4



수들의 합 2

161

<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

162

<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

163

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

- 찾으려고 하는 수: 5
- 합: 4



수들의 합 2

164

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

- 찾으려고 하는 수: 5
- 끝

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



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++;
    }
}
```

수들의 합 2

166

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

- 소스: <http://codeplus.codes/e3ba16a005da44ebbaffaab962291734>

부분합

167

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

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

부분합

168

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

- 소스: <http://codeplus.codes/23bef2cab780492d8b131223e6f5a745>

소수의 연속합

169

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

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

소수의 연속합

170

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

- 소스: <http://codeplus.codes/c0aeaaa0d142449191631c6ffe1126ba>

중간에서 만나기

중간에서 만나기

Meet in the Middle

172

- 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

173

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

- 서로 다른 N개의 정수로 이루어진 집합이 있을 때, 이 집합의 공집합이 아닌 부분집합 중에서 그 집합의 원소를 다 더한 값이 S가 되는 경우의 수를 구하는 문제
- $1 \leq N \leq 40$

부분집합의 합 2

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

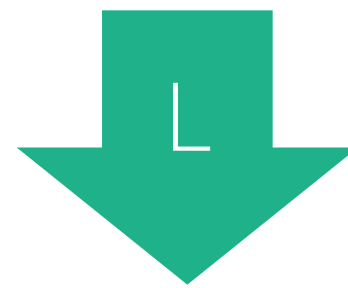
- 부분집합의 합 문제와 비슷하지만, 배열을 2개 써야하는 문제
- $A = [1, 2, 1, 3, 2, 1]$, $M = 4$ 인 경우를 생각해보자
- A 를 절반으로 나누어서
- $Up = [1, 2, 1]$
- $Down = [3, 2, 1]$
- 에 대해서 각각 모든 경우를 나열한다.

부분집합의 합 2

175

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

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



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

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



부분집합의 합 2

176

<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

177

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

- $0 + 5 = 5$ 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

178

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

- $0 + 4 = 4$ 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
---	---	---	---	---	---	---	---

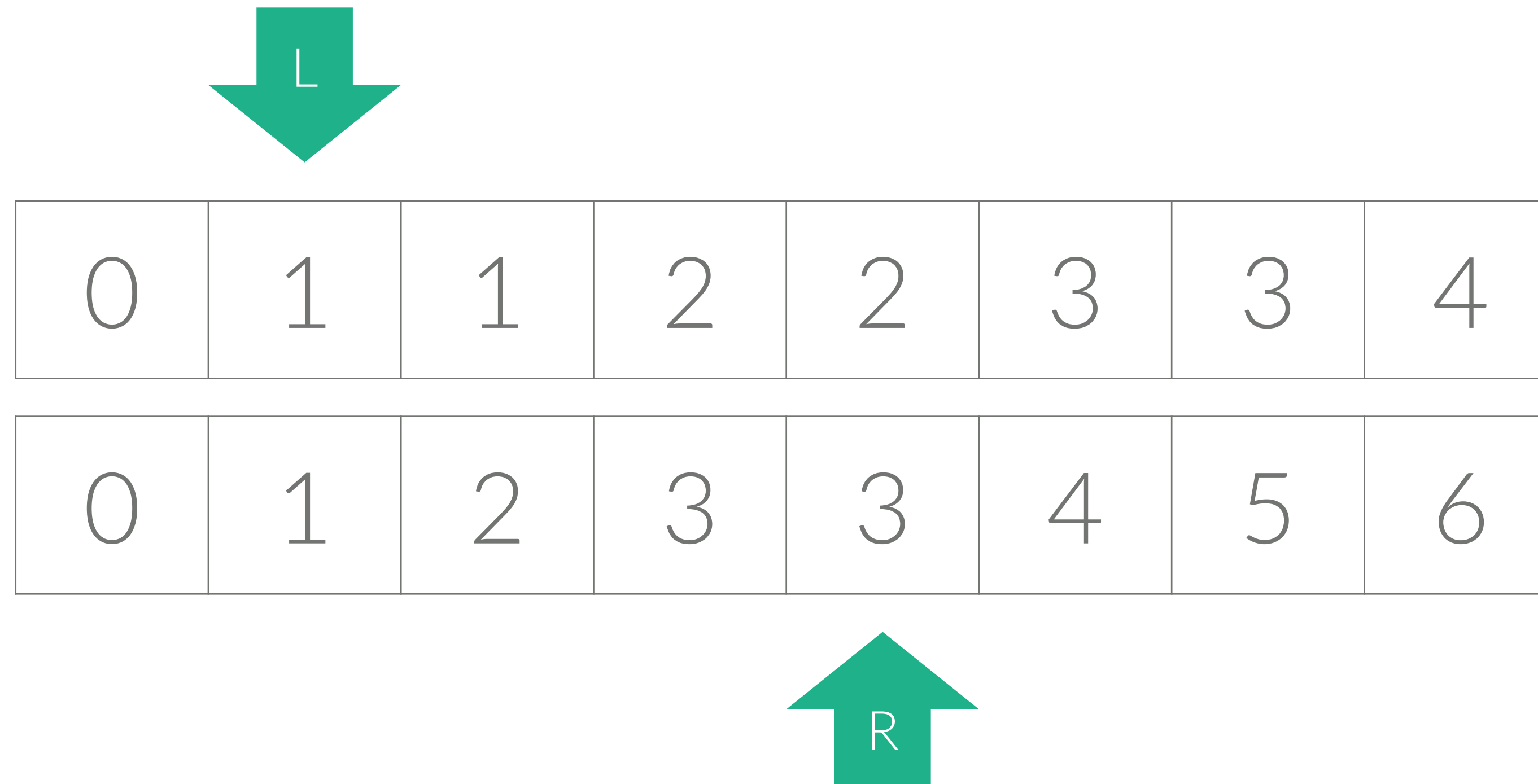


부분집합의 합 2

179

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

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

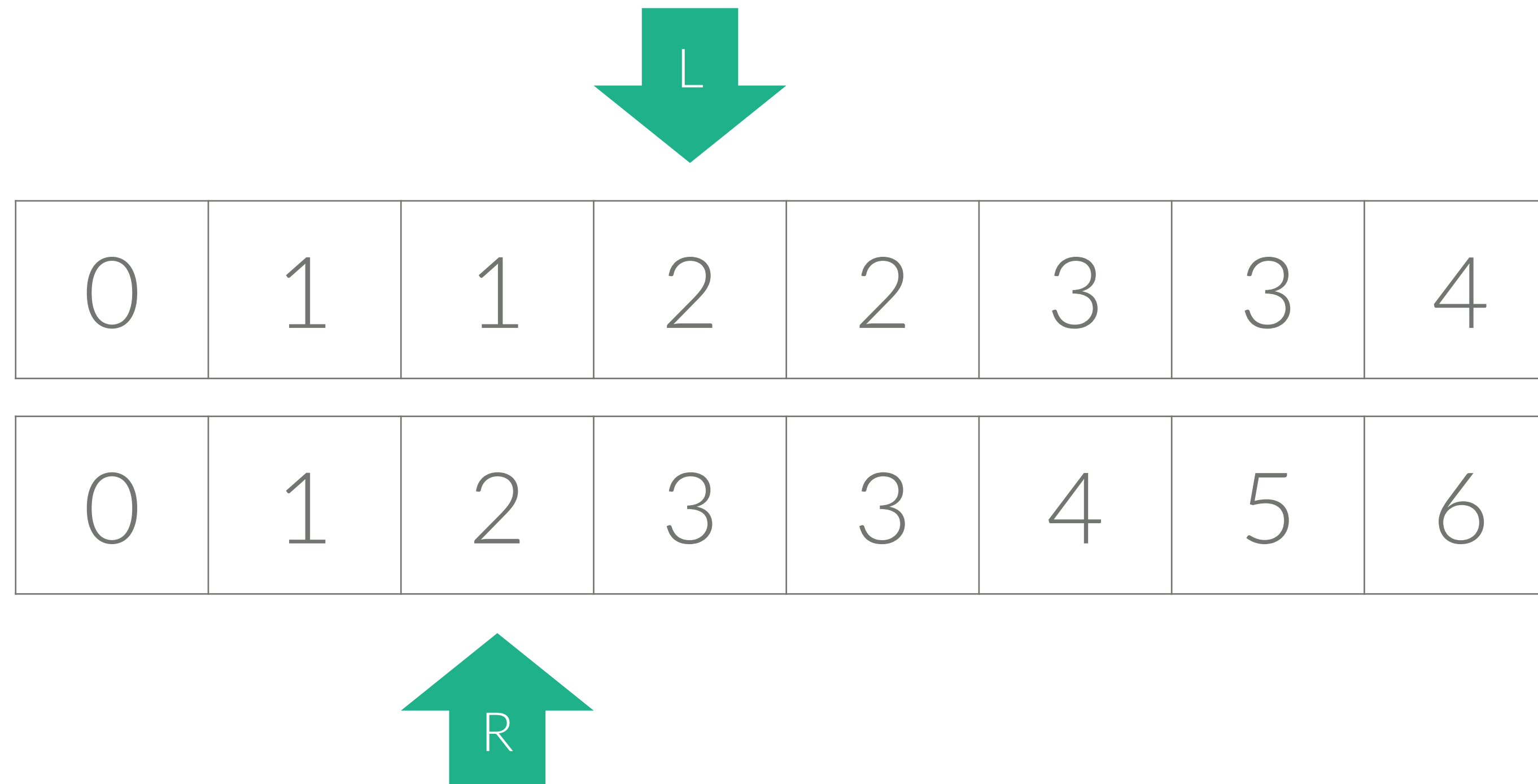


부분집합의 합 2

180

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

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

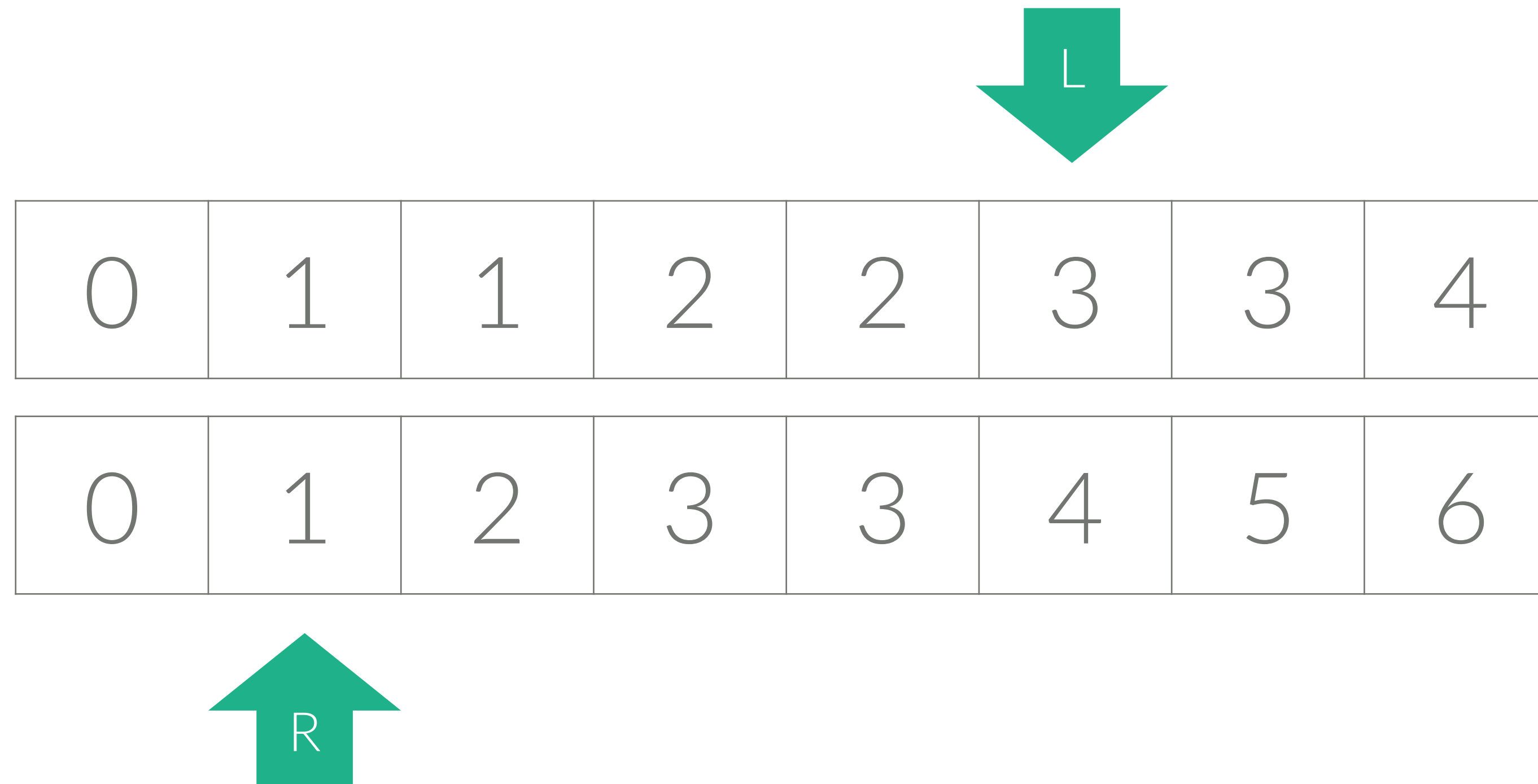


부분집합의 합 2

181

<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.

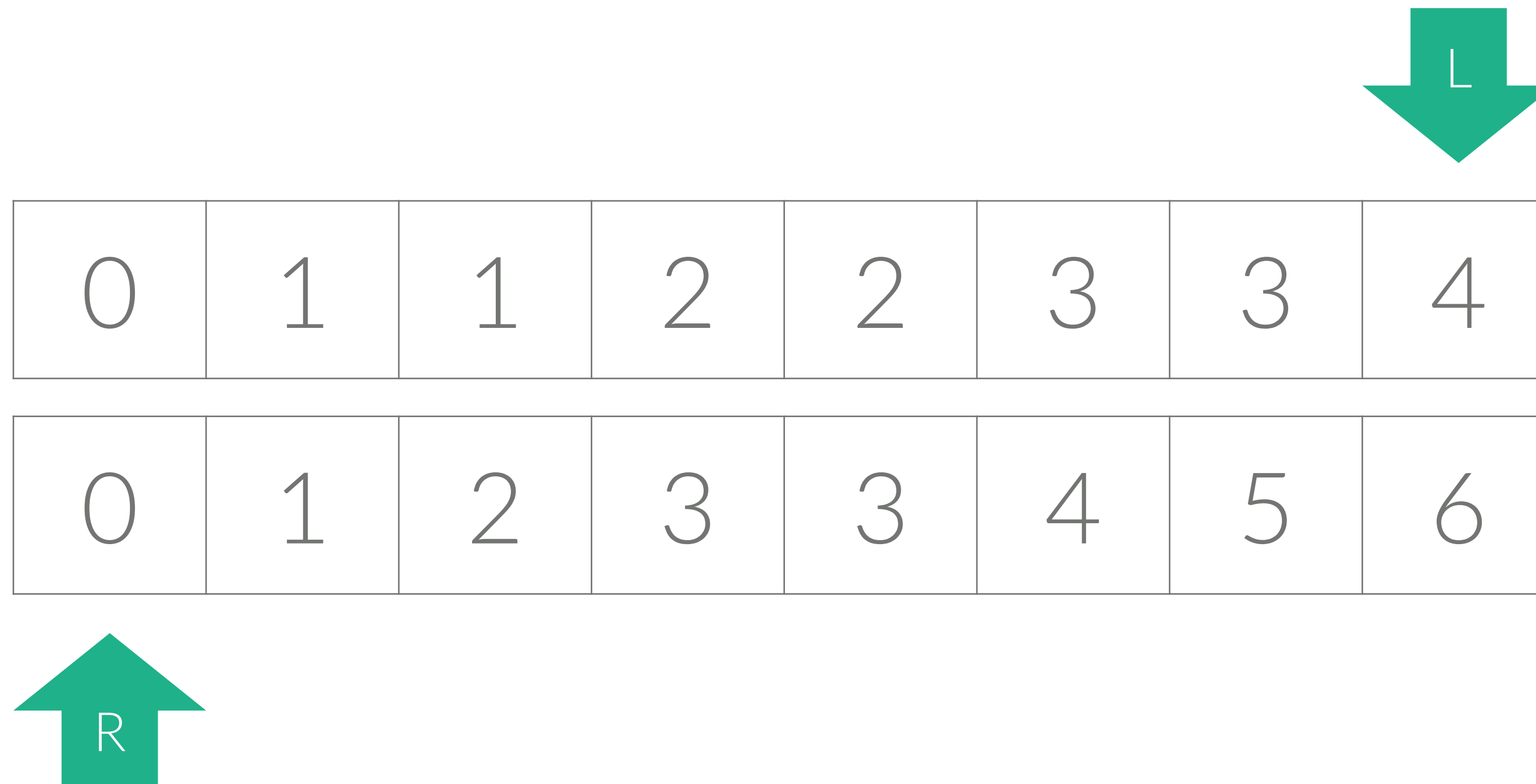


부분집합의 합 2

182

<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.



부분집합의 합 2

183

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

- 소스: <http://codeplus.codes/fd550951cfe54067a7b38c73039e6dc2>

두 배열의 합

<https://www.acmicpc.net/problem/2143>

- 배열 $A[1], A[2], \dots, A[n]$ 의 부 배열은 $A[i], A[i+1], \dots, A[j]$ ($1 \leq i \leq j \leq n$)의 합
- 두 배열 A와 B가 주어졌을 때
- A의 부 배열의 합과 B의 부 배열의 합을 더한 것이 T가 되는 경우의 수를 구하는 문제

두 배열의 합

185

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다



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

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

두 배열의 합

186

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $1 + B = T$, $B = 4$ 의 개수 (1개)



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

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

두 배열의 합

187

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $1 + B = T$, $B = 4$ 의 개수 (1개)



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

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

두 배열의 합

188

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $2 + B = T$, $B = 3$ 의 개수 (1개)



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

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

두 배열의 합

189

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $3 + B = T$, $B = 2$ 의 개수 (1개)



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

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

두 배열의 합

190

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $3 + B = T$, $B = 2$ 의 개수 (1개)



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

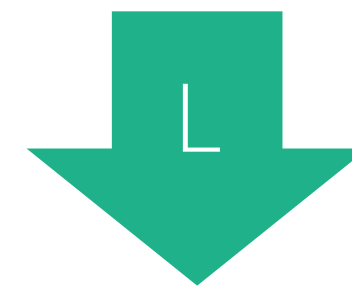
1	2	3	4	5	6
---	---	---	---	---	---

두 배열의 합

191

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $4 + B = T$, $B = 1$ 의 개수 (1개)



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

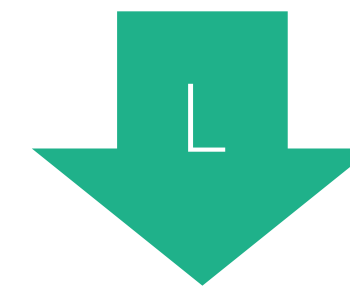
1	2	3	4	5	6
---	---	---	---	---	---

두 배열의 합

192

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $4 + B = T$, $B = 1$ 의 개수 (1개)



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

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

두 배열의 합

193

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $5 + B = T$, $B = 0$ 의 개수 (1개)



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

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

두 배열의 합

194

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $6 + B = T$, $B = -1$ 의 개수 (0개)



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

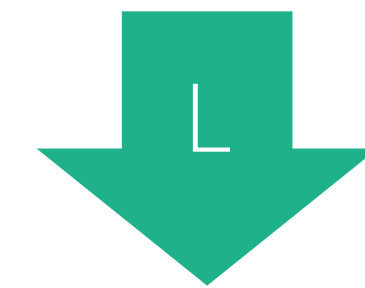
1	2	3	4	5	6
---	---	---	---	---	---

두 배열의 합

195

<https://www.acmicpc.net/problem/2143>

- $T = 5$, 각각의 A의 부분 합에 대해서, 해당하는 B의 부분 합의 개수를 세어보는 방법도 있다
- $7 + B = T$, $B = -2$ 의 개수 (0개)



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

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

두 배열의 합

196

<https://www.acmicpc.net/problem/2143>

- 소스: <http://codeplus.codes/4037cb2438514f868cc9ba3141249e30>

합이 0인 네 정수

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

- 크기가 N인 배열 A, B, C, D가 있다
- 이 때, $A[a]$, $B[b]$, $C[c]$, $D[d]$ 의 합이 0인 (a, b, c, d) 쌍의 개수를 구하는 문제
- $1 \leq N \leq 4,000$

합이 0인 네 정수

198

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

- 총 가능한 경우의 수: N^4 가지
- $A[a] + B[b] + C[c] + D[d] = 0$

합이 0인 네 정수

199

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

- 총 가능한 경우의 수: N^4 가지
- $A[a] + B[b] + C[c] + D[d] = 0$
- $A[a] + B[b] = -C[c] - D[d]$
- $A[a] + B[b] = -(C[c] + D[d])$

합이 0인 네 정수

200

<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

201

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

- 소스: <http://codeplus.codes/b3e45942e3774347af79b7af7df306d5>

끝

코드 플러스

203

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.