

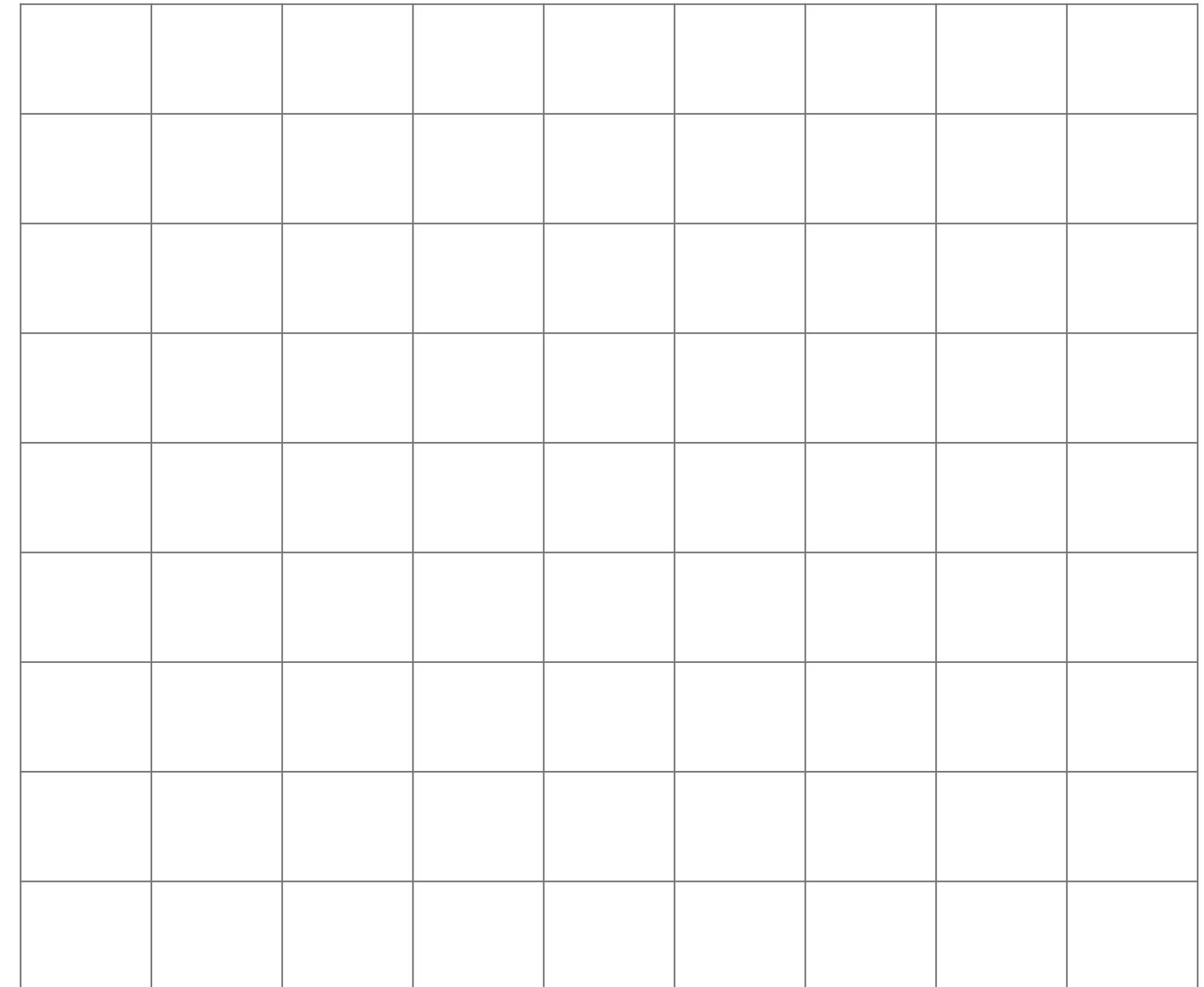
분할 정복 (연습)

최백준 choi@startlink.io

종이의 개수

<https://www.acmicpc.net/problem/1780>

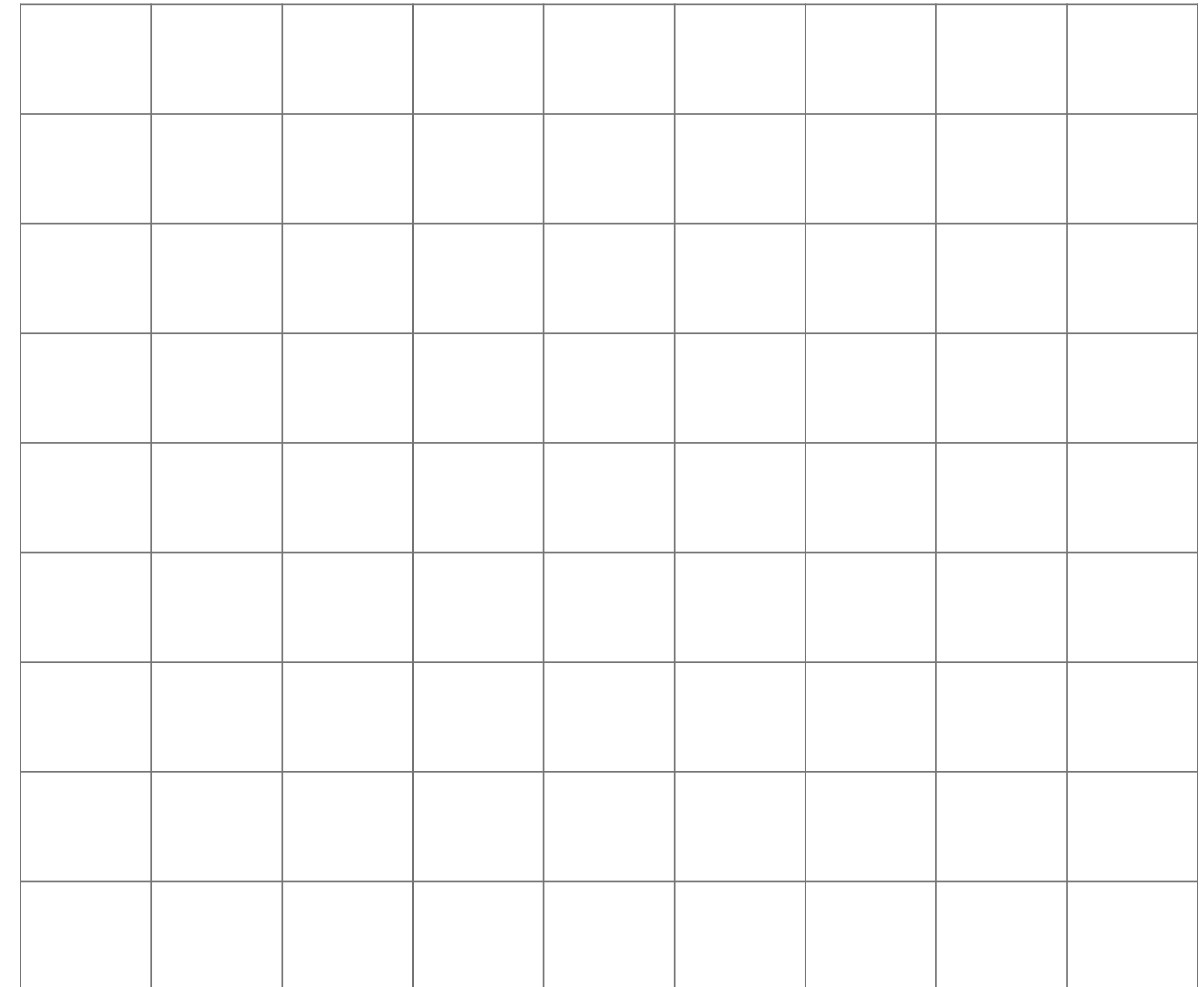
- 각 칸에 -1, 0, 1 중 하나가 써있는 $N \times N$ 크기의 행렬이 있다. $1 \leq N \leq 3^7$, $N = 3^k$ 꼴
- 종이가 같은 수이면 종이를 그대로, 그렇지 않은 경우에는 9등분을 하고 반복
- -1, 0, 1 종이의 개수를 구하는 문제



종이의 개수

<https://www.acmicpc.net/problem/1780>

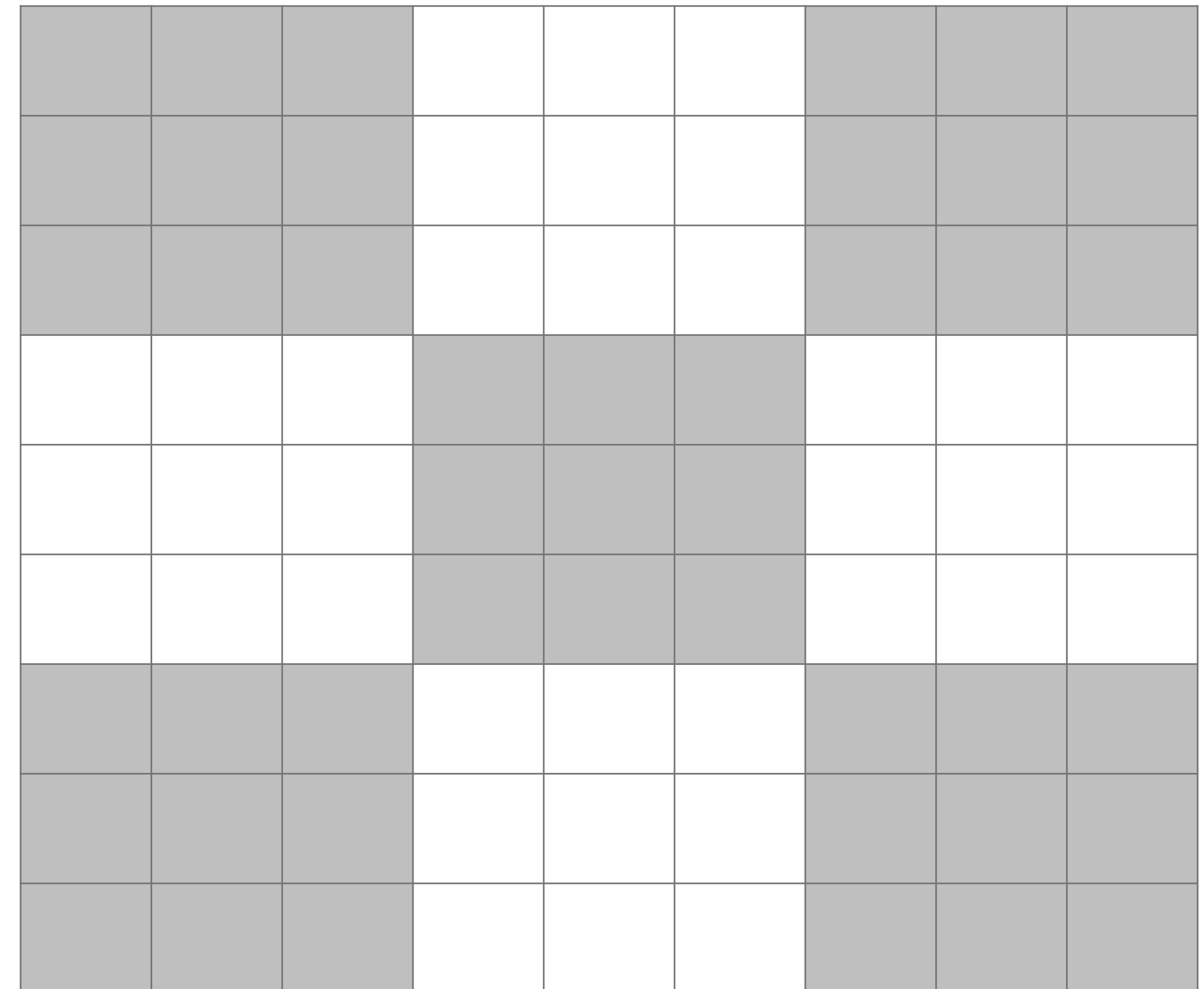
- `solve(x, y, n)`
- (x, y) 부터 가로로 n 개, 세로로 n 개의 종이 개수를 확인하는 함수



종이의 개수

<https://www.acmicpc.net/problem/1780>

- $\text{solve}(x, y, n)$
- (x, y) 부터 가로로 n 개, 세로로 n 개의 종이 개수를 확인하는 함수
- 작은 부분 문제는 총 9개
- $m = n/3$ 이라고 했을 때
- 부분 문제를 나눠보면



종이의 개수

<https://www.acmicpc.net/problem/1780>

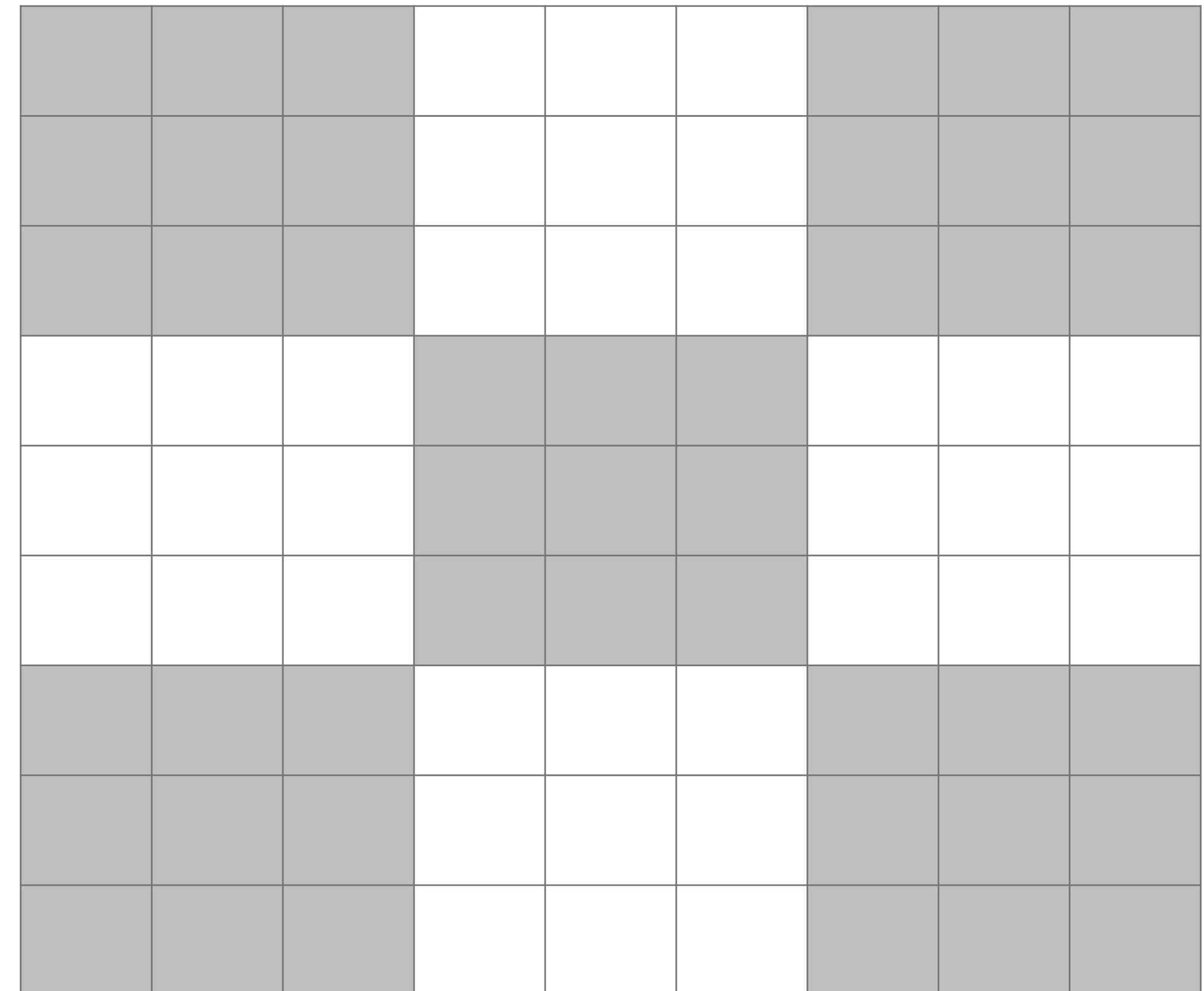
- $\text{solve}(x, y, n)$
 - 0: $\text{solve}(x, y, m)$
 - 1: $\text{solve}(x, y+m, m)$
 - 2: $\text{solve}(x, y+2*m, m)$
 - 3: $\text{solve}(x+m, y, m)$
 - 4: $\text{solve}(x+m, y+m, m)$
 - 5: $\text{solve}(x+m, y+2*m, m)$
 - 6: $\text{solve}(x+2*m, y, m)$
 - 7: $\text{solve}(x+2*m, y+m, m)$
 - 8: $\text{solve}(x+2*m, y+2*m, m)$

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

종이의 개수

<https://www.acmicpc.net/problem/1780>

- $\text{solve}(x, y, n)$
- (x, y) 부터 가로로 n 개, 세로로 n 개의 종이 개수를 확인하는 함수
- 부분 문제를 호출하기 전에
- 같은 수로 되어 있는지를 확인해야 한다
- 그게 아니면 부분 문제를 호출



종이의 개수

<https://www.acmicpc.net/problem/1780>

```
void solve(int x, int y, int n) {
    if (same(x, y, n)) {
        cnt[a[x][y]+1] += 1;
        return;
    }
    int m = n/3;
    for (int i=0; i<3; i++) {
        for (int j=0; j<3; j++) {
            solve(x+i*m, y+j*m, m);
        }
    }
}
```

종이의 개수

<https://www.acmicpc.net/problem/1780>

```
bool same(int x, int y, int n) {  
    for (int i=x; i<x+n; i++) {  
        for (int j=y; j<y+n; j++) {  
            if (a[x][y] != a[i][j]) {  
                return false;  
            }  
        }  
    }  
    return true;  
}
```


종이의 개수

<https://www.acmicpc.net/problem/1780>

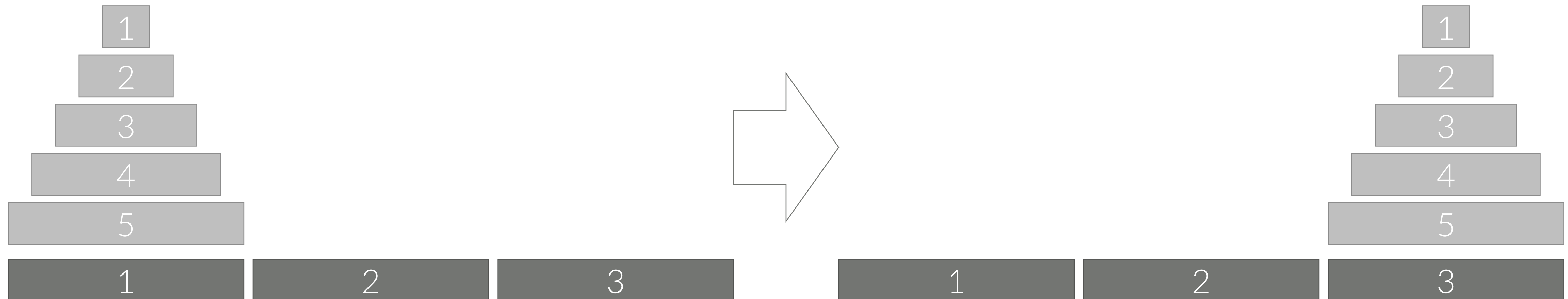
- 소스: <http://codeplus.codes/87eeb3c92ccc421bb733802134de4f2c>

하노이 탑 이동 순서

10

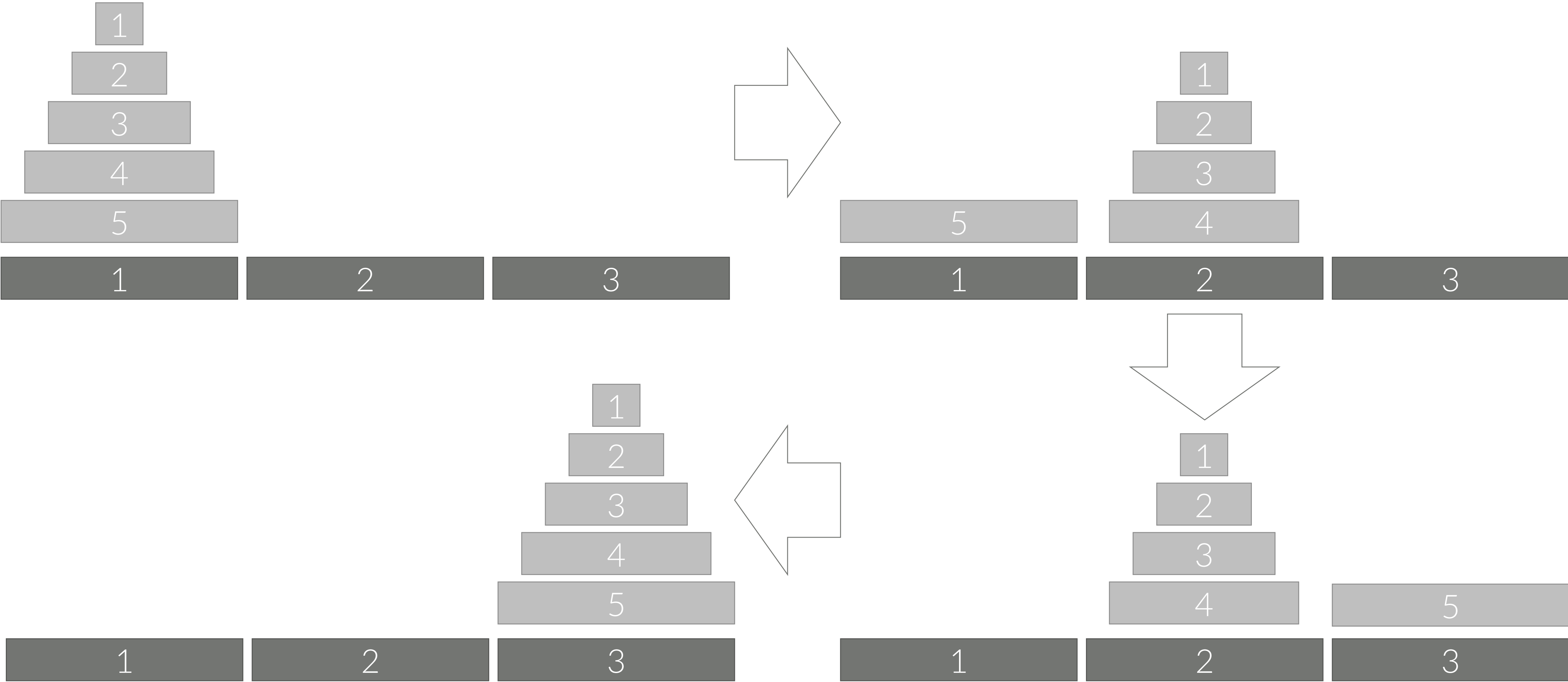
<https://www.acmicpc.net/problem/11729>

- 하노이 탑 문제
- 규칙 1: 한 번에 한 개의 원판만 다른 탑으로 옮길 수 있다
- 규칙 2: 쌓아 놓은 원판은 항상 위의 것이 아래의 것보다 작아야 한다 (중간 과정 포함)



하노이 탑 이동 순서

<https://www.acmicpc.net/problem/11729>



하노이 탑 이동 순서

<https://www.acmicpc.net/problem/11729>

- `solve(n, x, y)`
- 1~n개의 원판을 x에서 y로 이동하는 함수
- 1~n-1개의 원판을 x에서 z로 이동한다. (z는 x와 y가 아닌 원판)
- n번 원판을 x에서 y로 이동한다.
- 1~n-1개의 원판을 z에서 y로 이동한다.

하노이 탑 이동 순서

<https://www.acmicpc.net/problem/11729>

- `solve(n, x, y)`
- 1~n개의 원판을 x에서 y로 이동하는 함수
- 1~n-1개의 원판을 x에서 z로 이동한다. (z는 x와 y가 아닌 원판)
 - `solve(n-1, x, z)`
- n번 원판을 x에서 y로 이동한다.
- 1~n-1개의 원판을 z에서 y로 이동한다.
 - `solve(n-1, z, y)`

하노이 탑 이동 순서

<https://www.acmicpc.net/problem/11729>

```
void solve(int n, int x, int y) {  
    if (n == 0) return;  
    solve(n-1, x, 6-x-y);  
    printf("%d %d\n", x, y);  
    solve(n-1, 6-x-y, y);  
}
```

하노이 탑 이동 순서

15

<https://www.acmicpc.net/problem/11729>

- 소스: <http://codeplus.codes/55fd539ffe854b528651e44e89c5d9fd>

트리의 순회

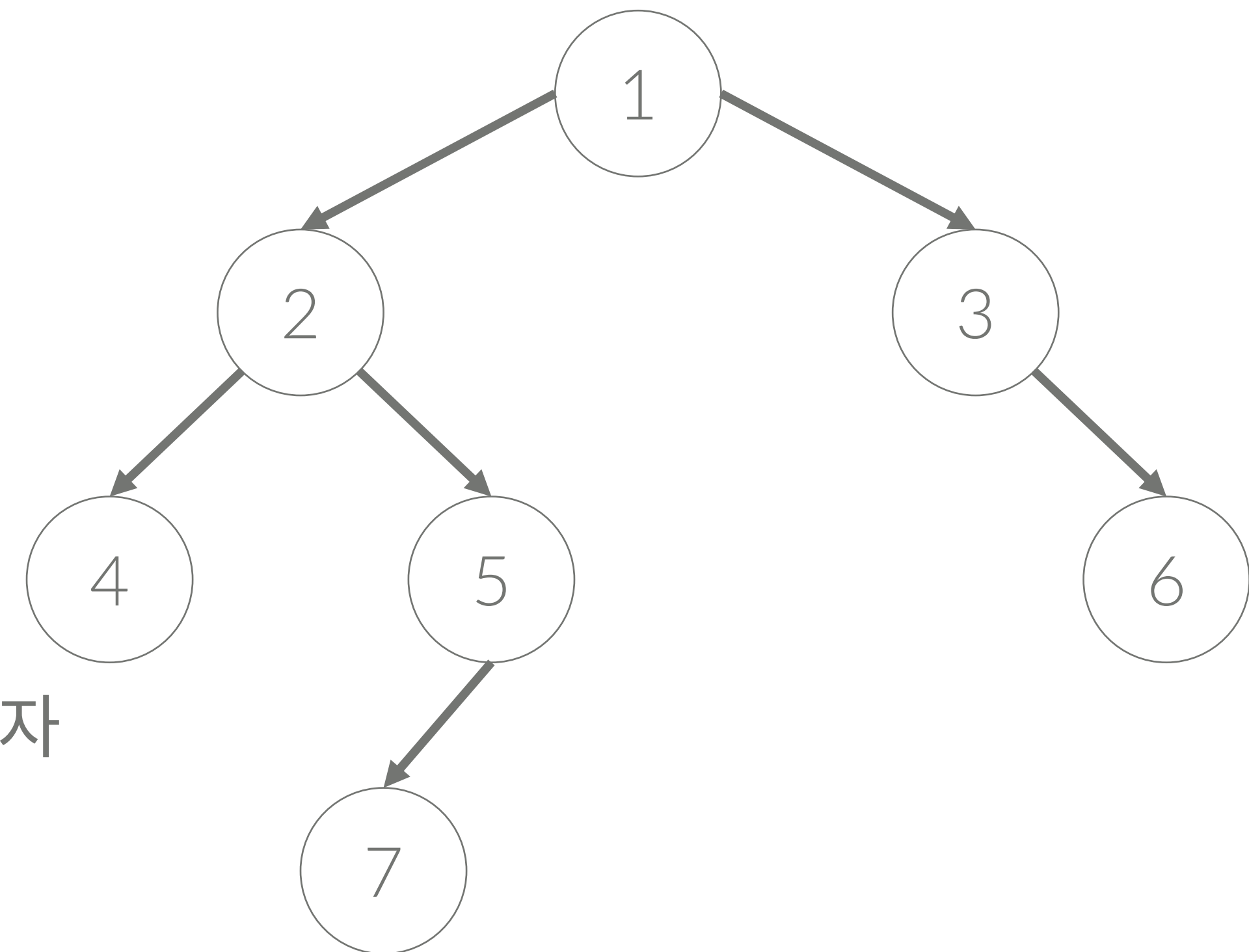
<https://www.acmicpc.net/problem/2263>

- N개의 정점을 갖는 이진 트리의 정점에 1부터 N까지 번호가 중복없이 매겨져 있다
- 이 이진 트리의 인오더와 포스트오더가 주어졌을 때 프리오더를 구하는 문제

트리의 순회

<https://www.acmicpc.net/problem/2263>

- 프리오더: 1 2 4 5 7 3 6
- 인오더: 4 2 7 5 1 3 6
- 포스트오더: 4 7 5 2 6 3 1
- 인오더와 포스트오더를 가지고 프리오더를 만들어 보자



트리의 순회

<https://www.acmicpc.net/problem/2263>

- 포스트오더의 가장 마지막은 루트이다.

인오더

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

포스트오더

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

트리의 순회

<https://www.acmicpc.net/problem/2263>

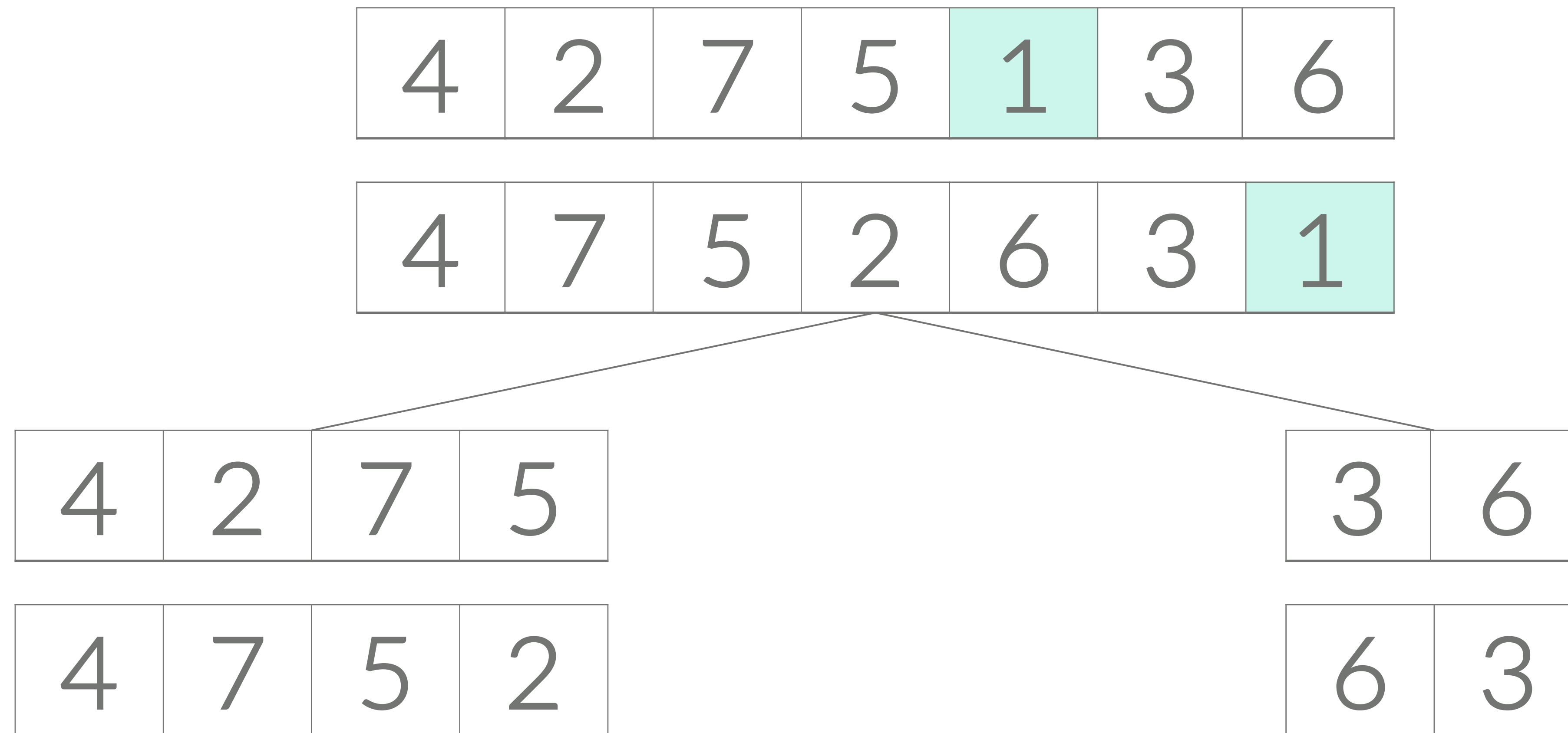
- 포스트오더의 가장 마지막은 루트이다.

	왼쪽	오른쪽
인오더	4275136	
포스트오더	4752631	
	왼쪽	오른쪽

트리의 순회

<https://www.acmicpc.net/problem/2263>

- 왼쪽과 오른쪽으로 나뉘서 풀 수 있다.



트리의 순회

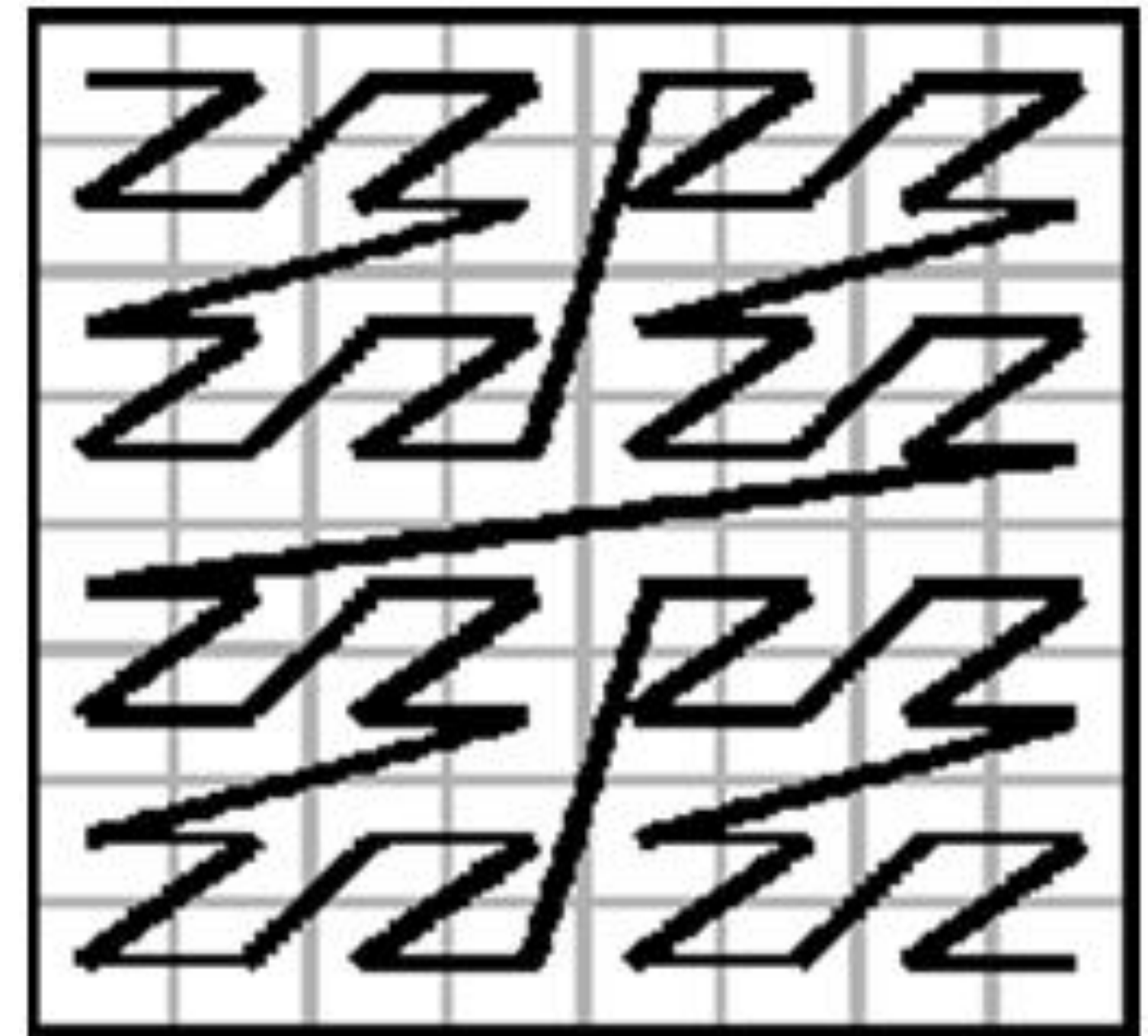
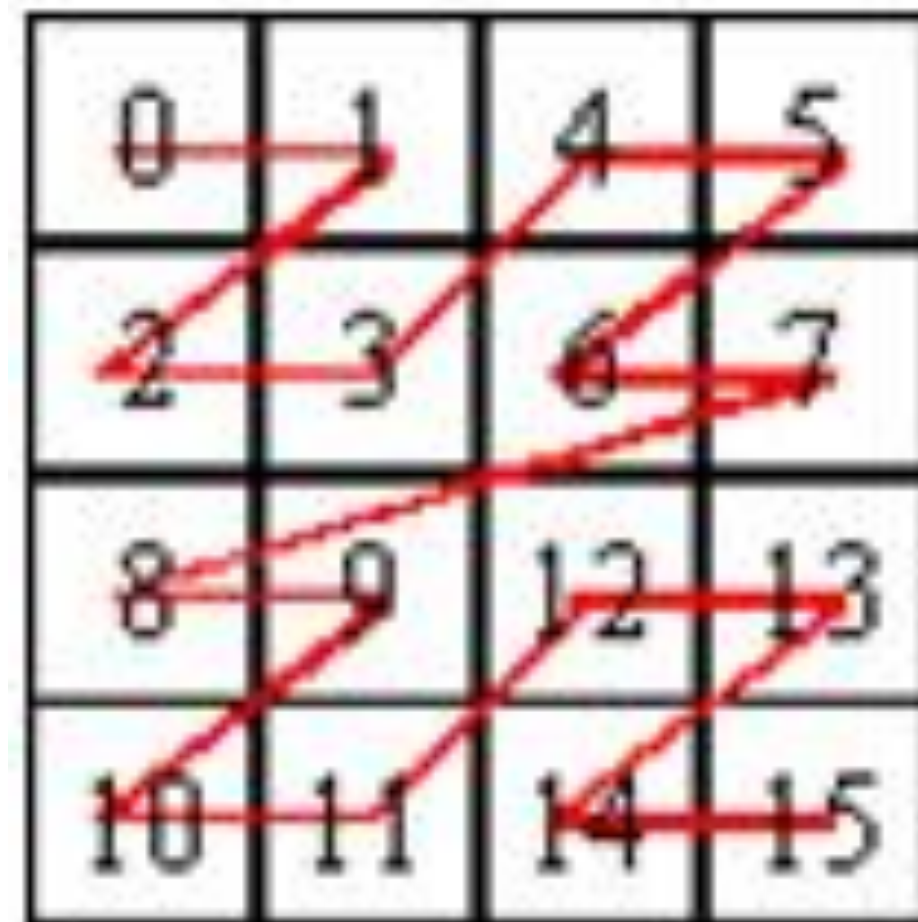
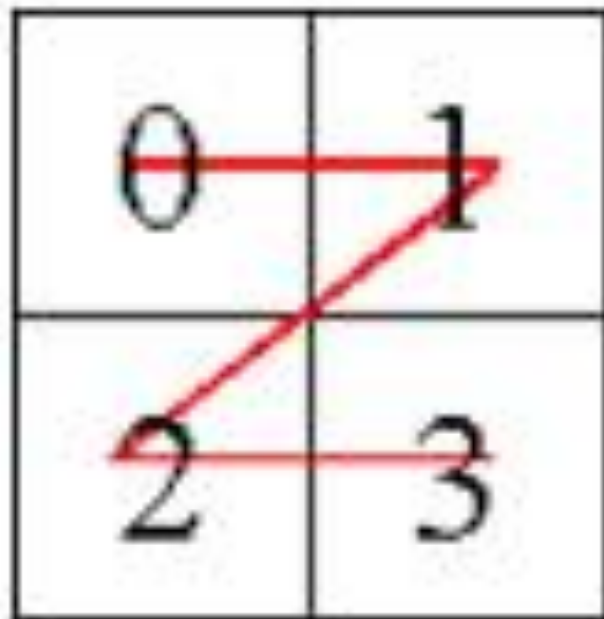
<https://www.acmicpc.net/problem/2263>

- 소스: <http://codeplus.codes/1d46d5b83862466b8f27b8e657c23eaf>

Z

<https://www.acmicpc.net/problem/1074>

- 만약, 2차원 배열의 크기가 $2^N * 2^N$ 라서 왼쪽 위에 있는 칸이 하나가 아니라면, 배열을 4등분한 후에 (크기가 같은 $2^{(N-1)}$ 로) 재귀적으로 순서대로 방문한다.



<https://www.acmicpc.net/problem/1074>

- 분할 정복을 이용해서 현재 칸이 4등분 했을 때, 어디에 속하는지 알아보고 분할 정복을 이용한다.

Z

<https://www.acmicpc.net/problem/1074>

- 소스: <http://codeplus.codes/3427a96ad06e4c3e9ae07806f3834698>

사분면

<https://www.acmicpc.net/problem/1891>

- 하나의 좌표평면은
- 네 개의 사분면으로 나뉜다



사분면

<https://www.acmicpc.net/problem/1891>

- 사분면은 다시 사분면으로 나뉜다

22	21	12	11
23	24	13	14
32	31	42	41
33	34	43	44

사분면

<https://www.acmicpc.net/problem/1891>

- 사분면 조각이 주어졌을 때
 - 입력으로 주어진 만큼 이동했을 때
 - 몇 번 조각으로 이동하는가?
-
- 341을 오른쪽으로 2칸, 위로 1칸 이동하면

222	221	212	211	122	121	112	111
223	224	213	214	123	124	113	114
232	231	242	241	132	131	142	141
233	234	243	244	133	134	143	144
322	321	312	311	422	421	412	411
323	324	313	314	423	424	413	414
332	331	342	341	432	431	442	441
333	334	343	344	433	434	443	444

사분면

<https://www.acmicpc.net/problem/1891>

- 사분면 조각이 주어졌을 때
 - 입력으로 주어진 만큼 이동했을 때
 - 몇 번 조각으로 이동하는가?
-
- 341을 오른쪽으로 2칸, 위로 1칸 이동하면
 - 424에 도착한다

222	221	212	211	122	121	112	111
223	224	213	214	123	124	113	114
232	231	242	241	132	131	142	141
233	234	243	244	133	134	143	144
322	321	312	311	422	421	412	411
323	324	313	314	423	424	413	414
332	331	342	341	432	431	442	441
333	334	343	344	433	434	443	444

사분면

<https://www.acmicpc.net/problem/1891>

- 두 개의 문제로 나누어져 있다고 볼 수 있다

사분면

<https://www.acmicpc.net/problem/1891>

- 두 개의 문제로 나누어져 있다고 볼 수 있다
 1. 어떤 사분면 조각의 좌표는 몇인가?
 2. 어떤 좌표의 사분면 조각은 무엇인가?

사분면

<https://www.acmicpc.net/problem/1891>

- 좌표를 매겨보자

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

사분면

<https://www.acmicpc.net/problem/1891>

- 341을 찾아보자

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

사분면

<https://www.acmicpc.net/problem/1891>

- 341을 찾아보자

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

사분면

<https://www.acmicpc.net/problem/1891>

- 341을 찾아보자

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

사분면

<https://www.acmicpc.net/problem/1891>

- `go(index, r, c, size)`
- index번째 글자를 찾아야 함
- 가장 왼쪽 위 칸은 r행 c열
- 변의 길이는 size

사분면

<https://www.acmicpc.net/problem/1891>

- `go(index, r, c, size)`
- 1번 사분면
 - `go(index, r, c+size/2, size/2)`
- 2번 사분면
- 3번 사분면
- 4번 사분면

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

사분면

<https://www.acmicpc.net/problem/1891>

- `go(index, r, c, size)`
- 1번 사분면
 - `go(index, r, c+size/2, size/2)`
- 2번 사분면
 - `go(index, r, c, size/2)`
- 3번 사분면
- 4번 사분면

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

사분면

<https://www.acmicpc.net/problem/1891>

- `go(index, r, c, size)`
- 1번 사분면
 - `go(index, r, c+size/2, size/2)`
- 2번 사분면
 - `go(index, r, c, size/2)`
- 3번 사분면
 - `go(index, r+size/2, c, size/2)`
- 4번 사분면

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

사분면

<https://www.acmicpc.net/problem/1891>

- `go(index, r, c, size)`
- 1번 사분면
 - `go(index, r, c+size/2, size/2)`
- 2번 사분면
 - `go(index, r, c, size/2)`
- 3번 사분면
 - `go(index, r+size/2, c, size/2)`
- 4번 사분면
 - `go(index, r+size/2, c+size/2, size/2)`

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

사분면

<https://www.acmicpc.net/problem/1891>

```
pair<long long, long long> go(int index, long long r, long long c,
long long size) {
    if (size == 1) {
        return make_pair(r, c);
    } else {
        if (a[index] == '1') {
            return go(index+1, r, c+size/2, size/2);
        } else if (a[index] == '2') {
            return go(index+1, r, c, size/2);
        } else if (a[index] == '3') {
            return go(index+1, r+size/2, c, size/2);
        } else if (a[index] == '4') {
            return go(index+1, r+size/2, c+size/2, size/2);
        }
    }
}
return make_pair(0, 0);
}
```


사분면

<https://www.acmicpc.net/problem/1891>

```
string gogo(long long r, long long c, long long size, long long x,
long long y) {
    if (size == 1) return "";
    if (x < r+size/2 && y < c+size/2) {
        return "2" + gogo(r, c, size/2, x, y);
    } else if (x < r+size/2 && y >= c+size/2) {
        return "1" + gogo(r, c+size/2, size/2, x, y);
    } else if (x >= r+size/2 && y < c+size/2) {
        return "3" + gogo(r+size/2, c, size/2, x, y);
    } else {
        return "4" + gogo(r+size/2, c+size/2, size/2, x, y);
    }
}
```

사분면

42

<https://www.acmicpc.net/problem/1891>

- 소스: <http://codeplus.codes/1bcf7d401af64bd685d52917efb964de>

별 찍기 - 10

<https://www.acmicpc.net/problem/2447>

- N = 27
- N은 항상 3^k ($0 \leq k \leq 8$)

```
*****
*  **  **  **  **  **  **  **  *
*****
***      *      *      ***
*  *      *  **  *      *  **  *  *
***      *      *      ***
*****
*  **  **  **  **  **  **  **  *
*****
*****
*  **  **  *      *  **  **  *
*****
***      ***      ***      ***
*  *      *  *      *  *      *  *
***      ***      ***      ***
*****
*  **  **  *      *  **  **  *
*****
*****
*  **  **  **  **  **  **  **  *
*****
***      *      *      ***
*  *      *  **  *      *  **  *
***      *      *      ***
*****
*  **  **  **  **  **  **  **  *
*****
```

별 찍기 - 10

<https://www.acmicpc.net/problem/2447>

- N = 27
- N은 항상 3^k ($0 \leq k \leq 8$)



별 찍기 - 10

45

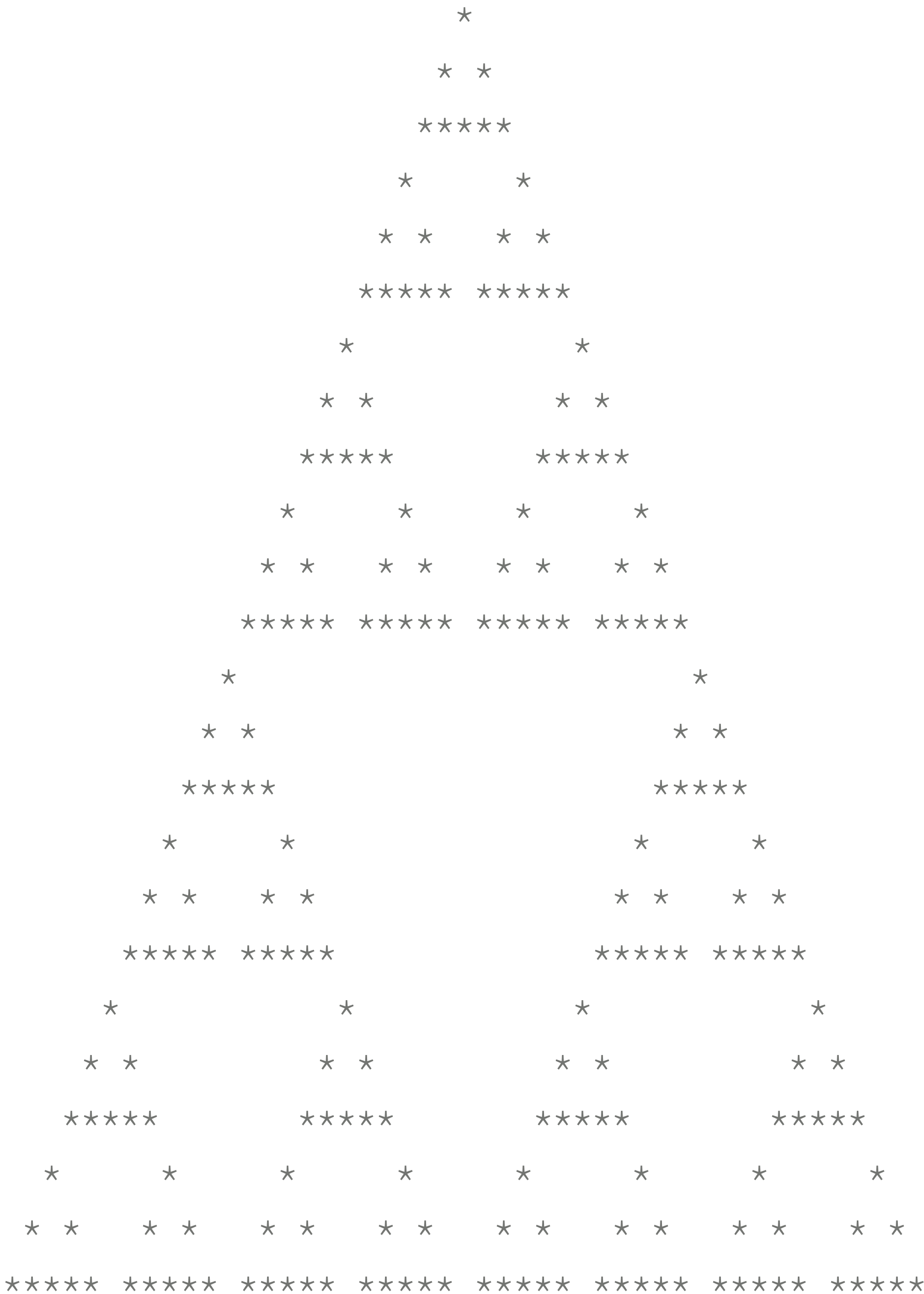
<https://www.acmicpc.net/problem/2447>

- 소스: <http://codeplus.codes/b89846a622724496b1819027335a2fb5>

별 찍기 - 11

<https://www.acmicpc.net/problem/2448>

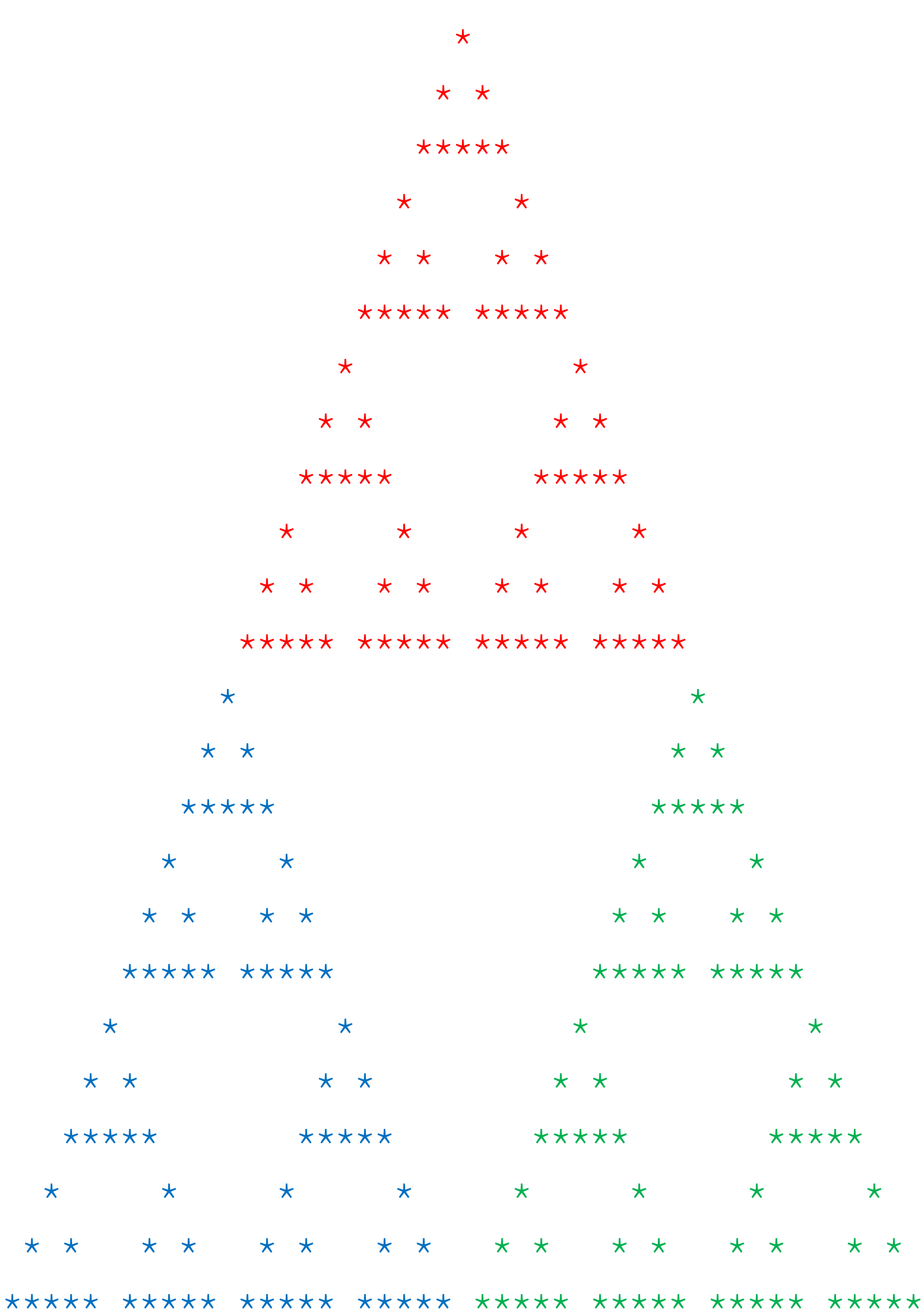
- N = 24
- N은 항상 3×2^k ($1 \leq k \leq 10$)



별 찍기 - 11

<https://www.acmicpc.net/problem/2448>

- N = 24
- N은 항상 3×2^k ($1 \leq k \leq 10$)



별 찍기 - 11

48

<https://www.acmicpc.net/problem/2448>

- 소스: <http://codeplus.codes/342b88874e514f9b93cac4b0bfc6bcb3>

버블 소트

<https://www.acmicpc.net/problem/1517>

- N개로 이루어진 수열 $A[1], A[2], \dots, A[N]$ 이 있을 때
- Swap이 총 몇 번 발생하는지 알아내는 문제
- $3\ 2\ 1 \rightarrow 2\ 3\ 1 \rightarrow 2\ 1\ 3 \rightarrow 1\ 2\ 3$ (3번)

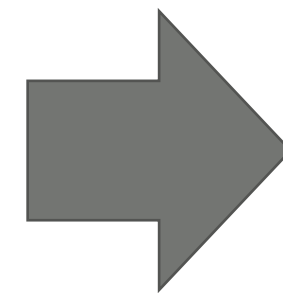
버블 소트

50

<https://www.acmicpc.net/problem/1517>

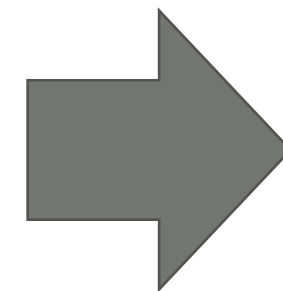
- $A = [10, 1, 5, 2, 3]$

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



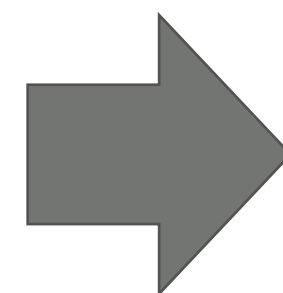
1	10	5	2	3
---	----	---	---	---

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



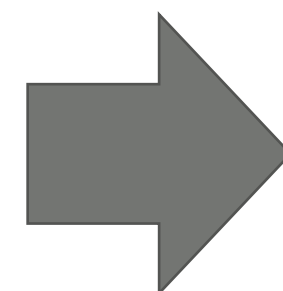
1	5	10	2	3
---	---	----	---	---

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



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

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



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

$i=1$

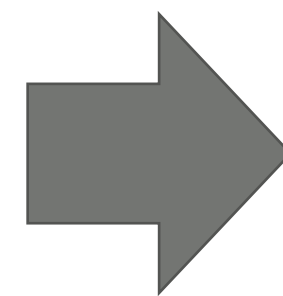
버블 소트

51

<https://www.acmicpc.net/problem/1517>

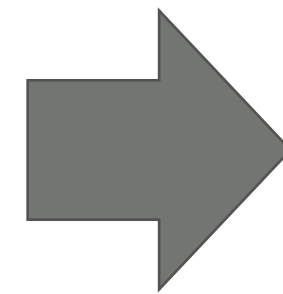
- $A = [10, 1, 5, 2, 3]$

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



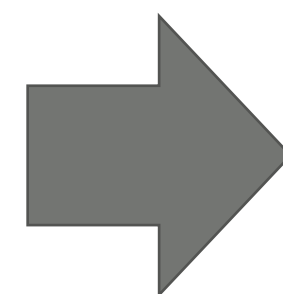
1	5	2	3	10
---	---	---	---	----

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



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

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



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

$i=2$

버블 소트

<https://www.acmicpc.net/problem/1517>

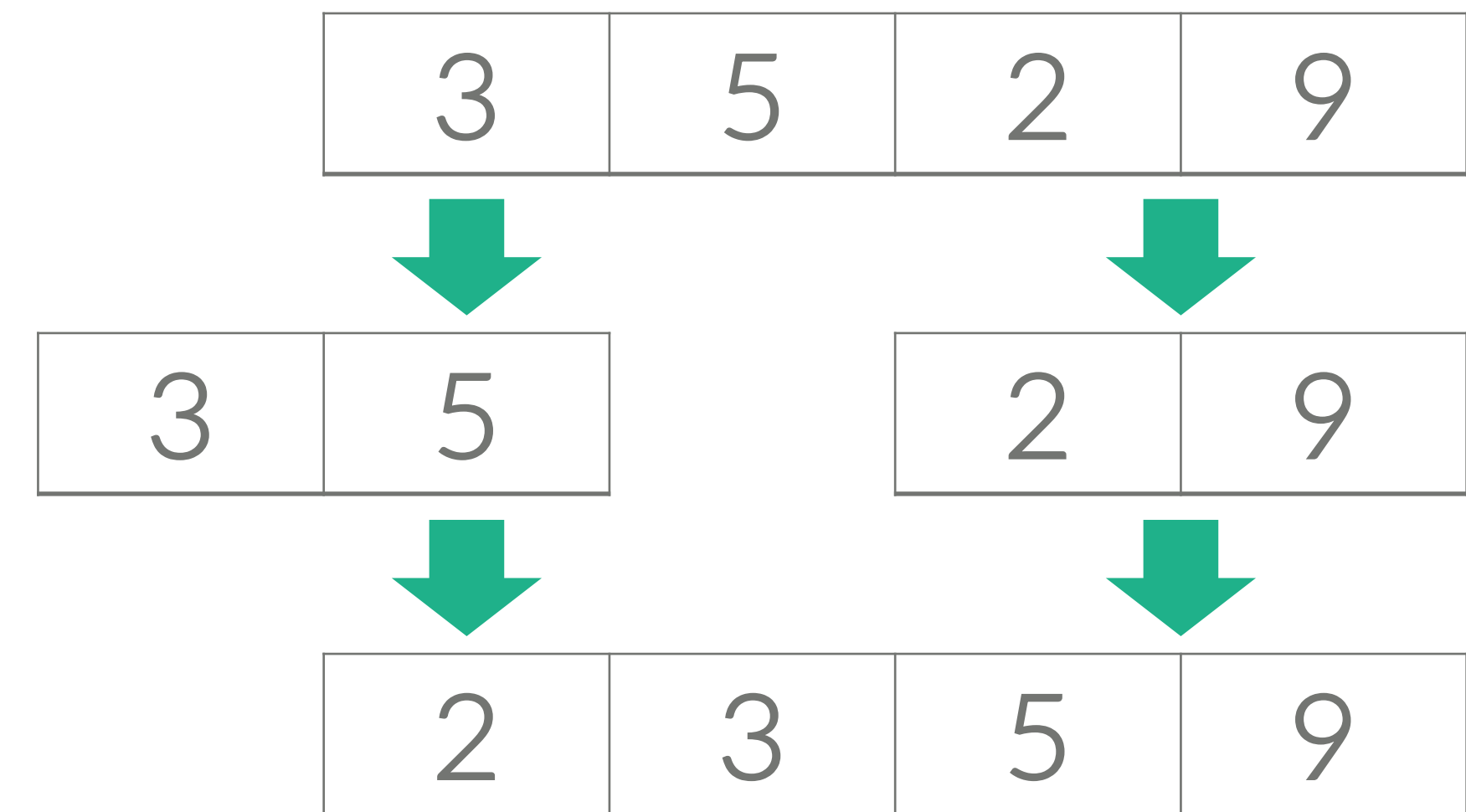
- 이 문제는 수열에서 inversion의 개수를 세는 문제이다.
- inversion: $A[i] > A[j]$ ($i < j$)
- 머지 소트를 하면서 문제를 풀 수 있다.

버블 소트

53

<https://www.acmicpc.net/problem/1517>

- 3 5 2 9의 inversion의 개수는
- $3 > 2$, $5 > 2$ 로 2개다.
- 오른쪽 절반이 이동할 때
- 왼쪽 절반에서 아직 이동하지 않은 것의 개수가
- 그 때의 inversion의 개수이다.
- 2가 먼저 이동하는데, 그 때 왼쪽 절반에는 3과 5가 있다.
- 이것은 $3 > 2$, $5 > 2$ 를 의미한다.

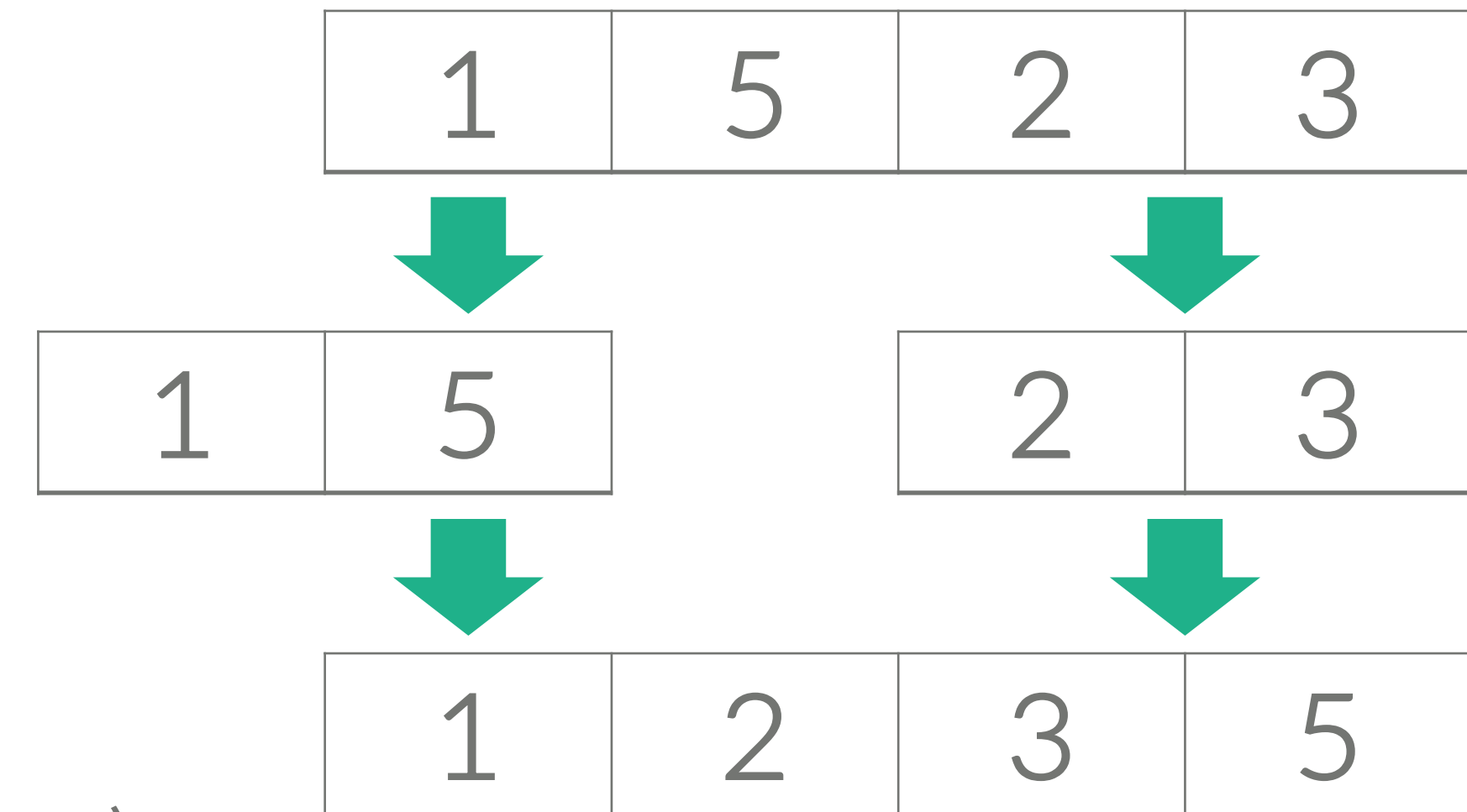


버블 소트

54

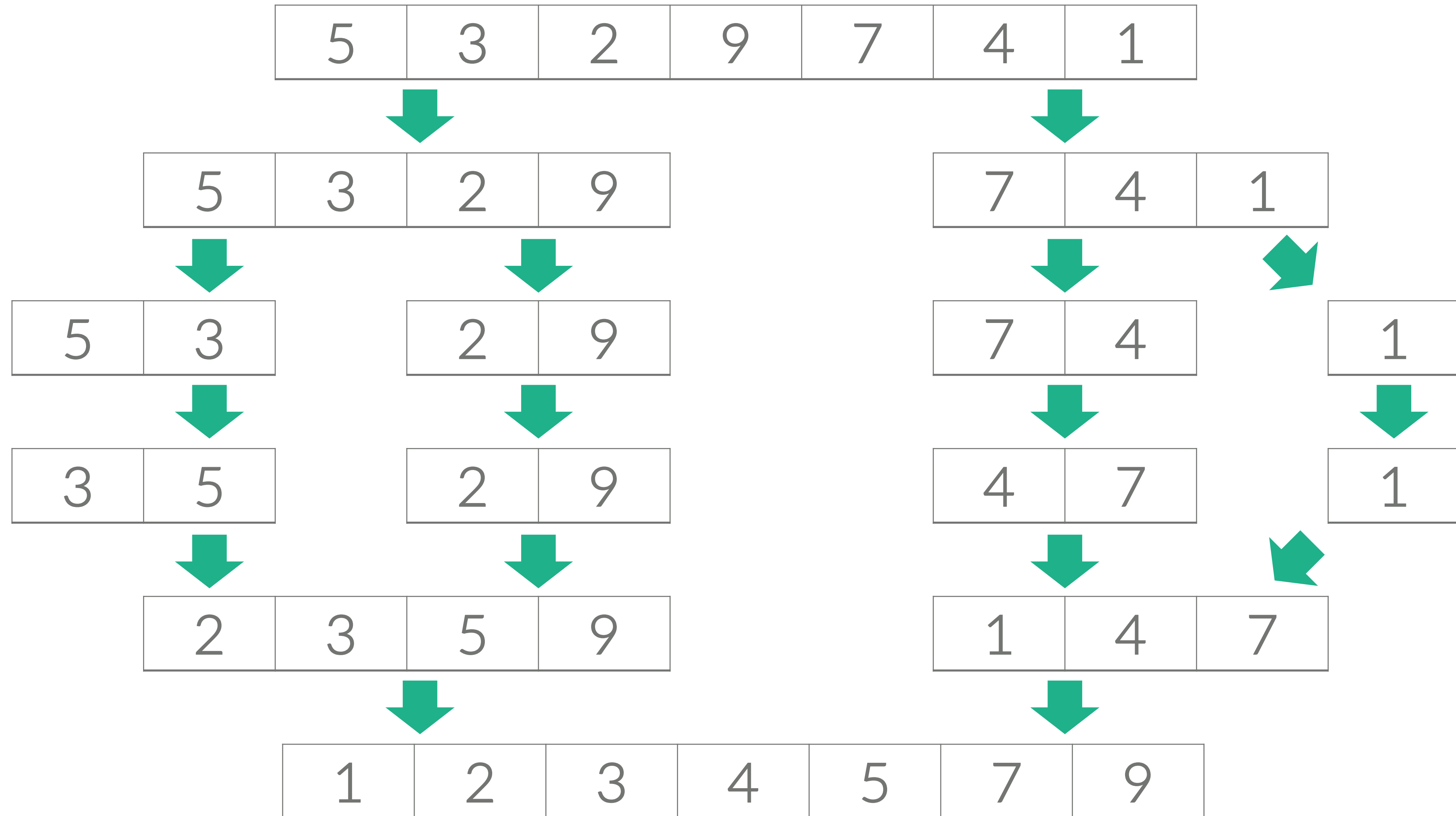
<https://www.acmicpc.net/problem/1517>

- 1 5 2 3의 inversion의 개수는
- $5 > 2, 5 > 3$ 로 2개다.
- 오른쪽 절반이 이동할 때
- 왼쪽 절반에서 아직 이동하지 않은 것의 개수가
- 그 때의 inversion의 개수이다.
- 2가 이동할 때, 왼쪽 절반에는 5가 남아있다. (inversion = 1)
- 3이 이동할 때도 왼쪽 절반에는 5가 남아있다. (inversion = 1)



버블 소트

<https://www.acmicpc.net/problem/1517>



버블 소트

<https://www.acmicpc.net/problem/1517>

```
long long solve(int start, int end) {  
    if (start == end) {  
        return 0;  
    }  
    int mid = (start+end)/2;  
    long long ans = solve(start, mid) + solve(mid+1, end);  
    int i = start;  
    int j = mid+1;  
    int k = 0;
```


버블 소트

<https://www.acmicpc.net/problem/1517>

```
while (i <= mid || j <= end) {
    if (i <= mid && (j > end || a[i] <= a[j])) {
        b[k++] = a[i++];
    } else {
        ans += (long long)(mid-i+1);
        b[k++] = a[j++];
    }
}
for (int i=start; i<=end; i++) {
    a[i] = b[i-start];
}
return ans;
}
```

버블 소트

<https://www.acmicpc.net/problem/1517>

- 소스: <http://codeplus.codes/fe697f6fb7224887bd7a5c17d2ff6399>