

브루트 포스

최백준 choi@startlink.io

브루트 포스

브루트 포스

Brute Force

3

- 브루트 포스는 모든 경우의 수를 다 해보는 것이다.

브루트 포스

Brute Force

- 예를 들어, 비밀번호가 4자리이고, 숫자로만 이루어져 있다고 한다면
- 0000부터 9999까지 다 입력해보면 된다.
- 경우의 수가 10,000가지 이다.

브루트 포스

Brute Force

- 예를 들어, 비밀번호가 4자리이고, 숫자로만 이루어져 있다고 한다면
- 0000부터 9999까지 다 입력해보면 된다.
- 경우의 수가 10,000가지 이다.
- 사람이 직접 비밀번호를 입력하는데 1초가 걸린다면 $10,000\text{초} = 2.7\text{시간}$ 정도 걸린다.

브루트 포스

Brute Force

- 예를 들어, 비밀번호가 12자리이고, 숫자로만 이루어져 있다고 한다면
- 000000000000부터 999999999999까지 다 입력해보면 된다.
- 경우의 수가 1,000,000,000,000가지 이다.

$$10^{12}$$

브루트 포스

Brute Force

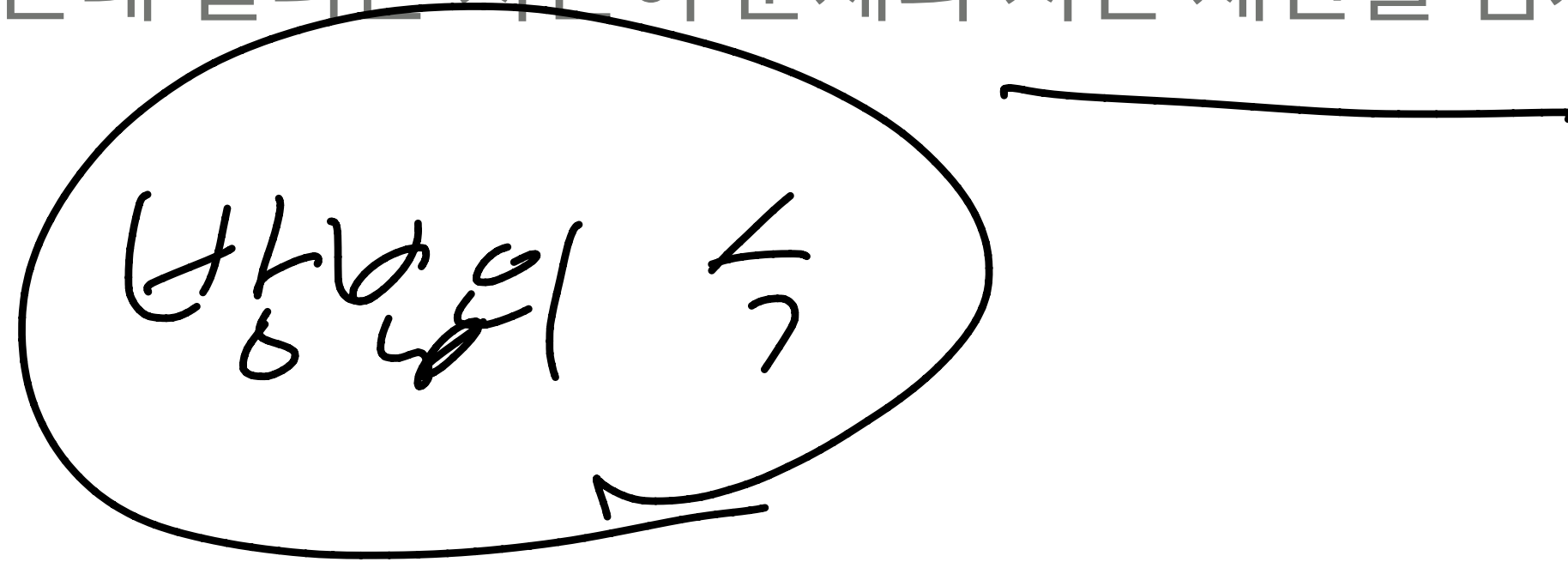
7

- 예를 들어, 비밀번호가 12자리이고, 숫자로만 이루어져 있다고 한다면
- 000000000000부터 999999999999까지 다 입력해보면 된다.
- 경우의 수가 1,000,000,000,000가지 이다.
- 사람이 직접 비밀번호를 입력하는데 1초가 걸린다면 1,000,000,000,000초 = 약 31688년이 걸린다.

브루트 포스

Brute Force

- 브루트 포스는 모든 경우의 수를 다 해보는 것이다.
- 이 때, 경우의 수를 다 해보는데 걸리는 시간이 문제의 시간 제한을 넘지 않아야 한다.



브루트 포스

Brute Force

- 브루트 포스로 문제를 풀기 위해서는 다음과 같은 3가지 단계를 생각해볼 수 있다.

1. 문제의 가능한 경우의 수를 계산해본다.

2. 가능한 모든 방법을 다 만들어본다.

3. 각각의 방법을 이용해 답을 구해본다.

브루트 포스

Brute Force

- 브루트 포스로 문제를 풀기 위해서는 다음과 같은 3가지 단계를 생각해볼 수 있다.
 1. 문제의 가능한 경우의 수를 계산해본다.
 - 직접 계산을 통해서 구한다. 대부분 손으로 계산해볼 수 있다.
 2. 가능한 모든 방법을 다 만들어본다.
 3. 각각의 방법을 이용해 답을 구해본다.

브루트 포스

Brute Force

- 브루트 포스로 문제를 풀기 위해서는 다음과 같은 3가지 단계를 생각해볼 수 있다.
1. 문제의 가능한 경우의 수를 계산해본다.
 - 직접 계산을 통해서 구한다. 대부분 손으로 계산해볼 수 있다.
 2. 가능한 모든 방법을 다 만들어본다.
 - 하나도 빠짐 없이 만들어야 한다.
 - 대표적으로 그냥 다 해보는 방법, for문 사용, 순열 사용, 재귀 호출 사용, 비트마스크 사용이 있다.
 3. 각각의 방법을 이용해 답을 구해본다.

브루트 포스

Brute Force

- 브루트 포스로 문제를 풀기 위해서는 다음과 같은 3가지 단계를 생각해볼 수 있다.
1. 문제의 가능한 경우의 수를 계산해본다.
 - 직접 계산을 통해서 구한다. 대부분 손으로 계산해볼 수 있다.
 2. 가능한 모든 방법을 다 만들어본다.
 - 하나도 빠짐 없이 만들어야 한다.
 - 대표적으로 그냥 다 해보는 방법, for문 사용, 순열 사용, 재귀 호출 사용, 비트마스크 사용이 있다.
 3. 각각의 방법을 이용해 답을 구해본다.
 - 이 단계는 보통은 어렵지 않다. 문제에 나와있는 대로 답을 계산해본다.

브루트 포스

Brute Force

① 방한의 수 × 방법 (개수)
시간복잡도

13

- 브루트 포스로 문제를 풀기 위해서는 다음과 같은 3가지 단계를 생각해볼 수 있다.
- 1. 문제의 가능한 경우의 수를 계산해본다.
 - 직접 계산을 통해서 구한다. 대부분 손으로 계산해볼 수 있다.
- 2. 가능한 모든 방법을 다 만들어본다.
 - 하나도 빠짐 없이 만들어야 한다.
 - 대표적으로 그냥 다 해보는 방법, for문 사용, 순열 사용, 재귀 호출 사용, 비트마스크 사용이 있다.
- 3. 각각의 방법을 이용해 답을 구해본다.
 - 이 단계는 보통은 어렵지 않다. 문제에 나와있는 대로 답을 계산해본다.
- 브루트 포스 문제의 시간 복잡도는 대부분 $O(\text{경우의 수} * \text{방법 1개를 시도해보는데 걸리는 시간 복잡도})$ 가 걸린다.

경우의 수

경우의 수

Brute Force

15

- 문제의 가능한 경우의 수를 계산하기 위해 몇 가지 경우의 수를 계산하는 방법을 연습해보자

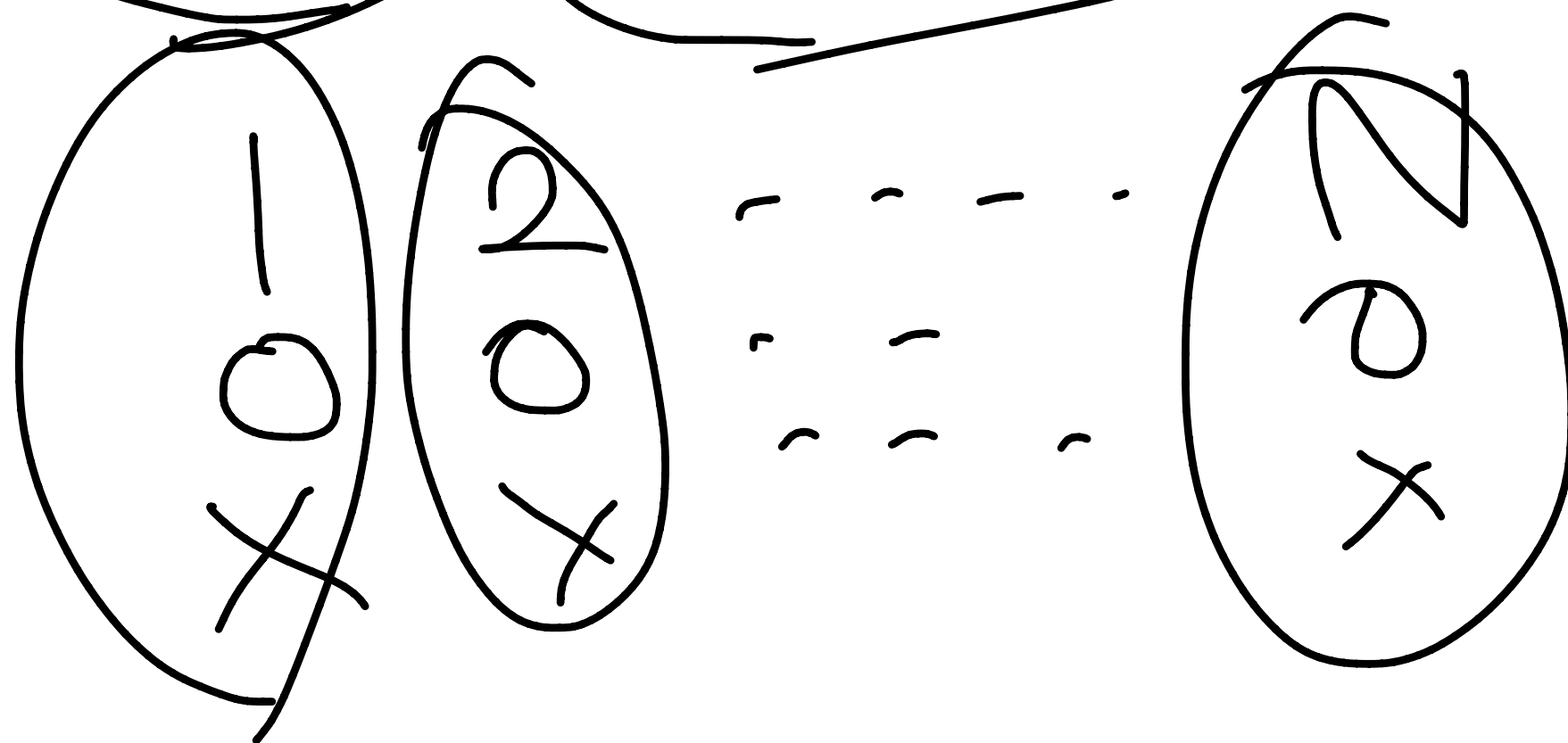
경우의 수

Brute Force

1, 2, 3, ..., (N)

$$\frac{N}{1} \times \frac{N-1}{2} \times \frac{N-2}{3} \times \dots \times \frac{1}{N}$$

- N명의 사람이 한 줄로 서는 경우의 수 $\rightarrow N!$
- N명의 사람 중에서 대표 두 명을 뽑는 경우의 수 $\rightarrow N C_2 = \frac{N \cdot (N-1)}{2}$
- N명의 사람 중에서 대표 세 명을 뽑는 경우의 수 $\rightarrow N C_3 = \frac{N \cdot (N-1) \cdot (N-2)}{3!}$
- N명의 사람 중에서 반장 1명과 부반장 1명을 뽑는 경우의 수 $\rightarrow N \times (N-1)$
- N명의 사람이 있을 때, 각 사람이 영화를 볼지, 보지 않을지 결정한다. 가능한 조합의 수 $\rightarrow 2^N$



경우의 수

Brute Force

- N명의 사람이 한 줄로 서는 경우의 수 $\rightarrow N \times (N-1) \times \dots \times 1 = N!$ $O(N!)$
- N명의 사람 중에서 대표 두 명을 뽑는 경우의 수 $\rightarrow N \times (N-1) / 2$ $\Theta(N^2)$
- N명의 사람 중에서 대표 세 명을 뽑는 경우의 수 $\rightarrow N \times (N-1) \times (N-2) / 3!$ $\frac{\Theta(N^3)}{O(N^2)}$
- N명의 사람 중에서 반장 1명과 부반장 1명을 뽑는 경우의 수 $\rightarrow N \times (N-1)$ $O(N^2)$
- N명의 사람이 있을 때, 각 사람이 영화를 볼지, 보지 않을지 결정한다. 가능한 조합의 수 $\rightarrow 2^N$ $O(2^N)$

그냥 다 해보기

일곱 난쟁이

<https://www.acmicpc.net/problem/2309>

- 아홉 명의 난쟁이 중 일곱 명의 난쟁이를 찾는 문제
- 일곱 난쟁이의 키의 합은 100이다.

난쟁이 키

$$q_2 = \frac{9 \times 8^4}{2 \times 1} = 36$$

일곱 난쟁이

20

<https://www.acmicpc.net/problem/2309>

- 아홉 명 중에 일곱 명을 고르는 것은
- 아홉 명 중에 두 명을 고르는 것과 같다.

일곱 난쟁이

<https://www.acmicpc.net/problem/2309>

- 아홉 명 중에 일곱 명을 고르는 것은
- 아홉 명 중에 두 명을 고르는 것과 같다.
- 난쟁이의 수를 N 이라고 했을 때
- 두 명을 고르는 경우의 수: N^2 라고 할 수 있다.
- 나머지 난쟁이의 키의 합을 고르는 시간 복잡도: $O(N)$

일곱 난쟁이

<https://www.acmicpc.net/problem/2309>

- 아홉 명 중에 일곱 명을 고르는 것은
- 아홉 명 중에 두 명을 고르는 것과 같다.
- 난쟁이의 수를 N 이라고 했을 때
- 두 명을 고르는 경우의 수: N^2 라고 할 수 있다.
- 나머지 난쟁이의 키의 합을 고르는 시간 복잡도: $O(N)$
- 따라서, 이 문제는 $O(N^3)$ 으로 해결할 수 있다.

일곱 난쟁이

23

<https://www.acmicpc.net/problem/2309>

- 소스: <http://codeplus.codes/4d5ea4a198e0451cb6c6432825a24b34>

사탕 게임

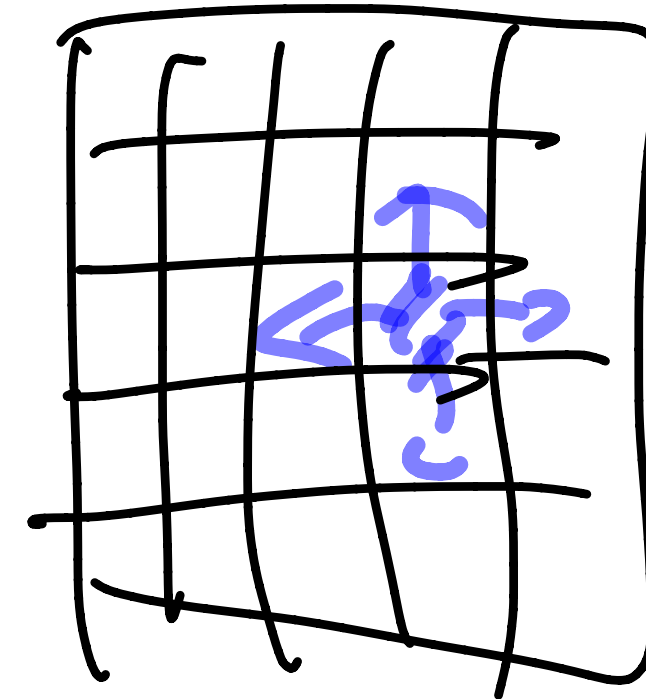
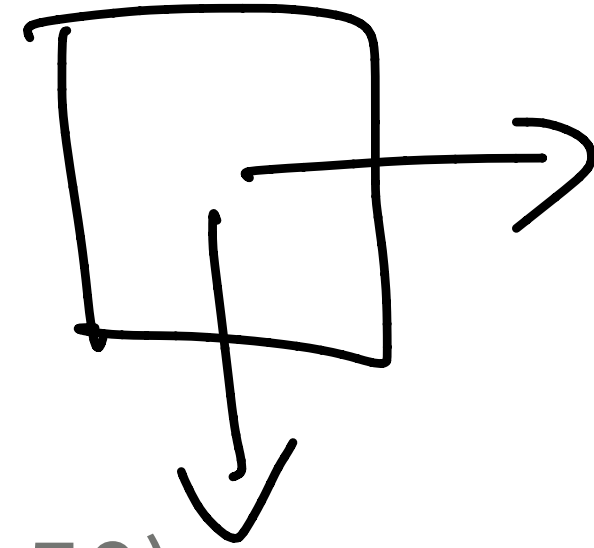
<https://www.acmicpc.net/problem/3085>

24

- $N \times N$ 크기의 테이블에 사탕이 있다. ($N \leq 50$)

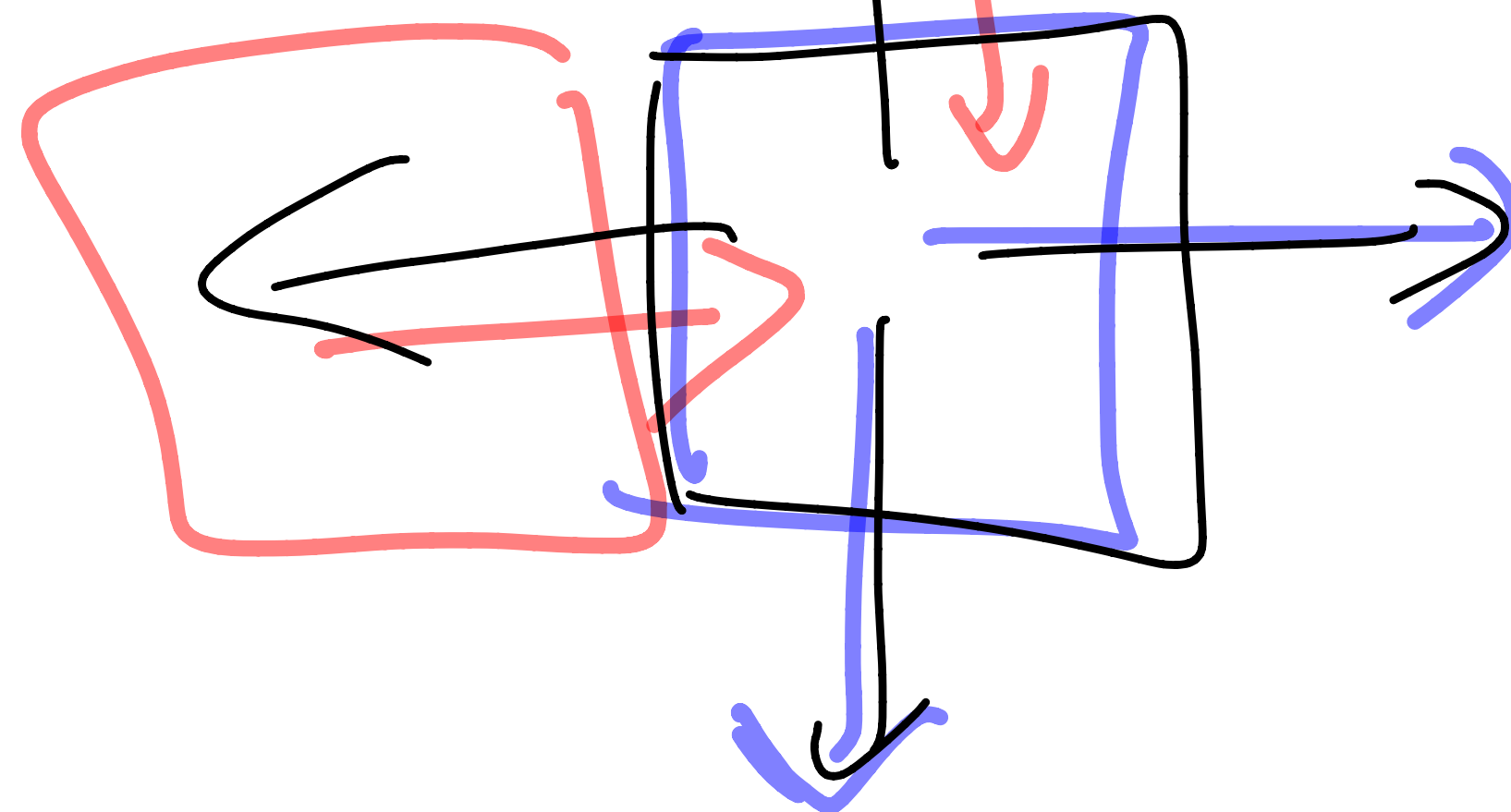
- 인접한 두 칸을 고르고, 사탕을 교환한다.

- 그 다음, 같은 색으로 이루어져 있는 가장 긴 연속 부분 행 또는 열을 고르는 문제



칸의 개수: N^2

방향의 수: $O(N^2)$



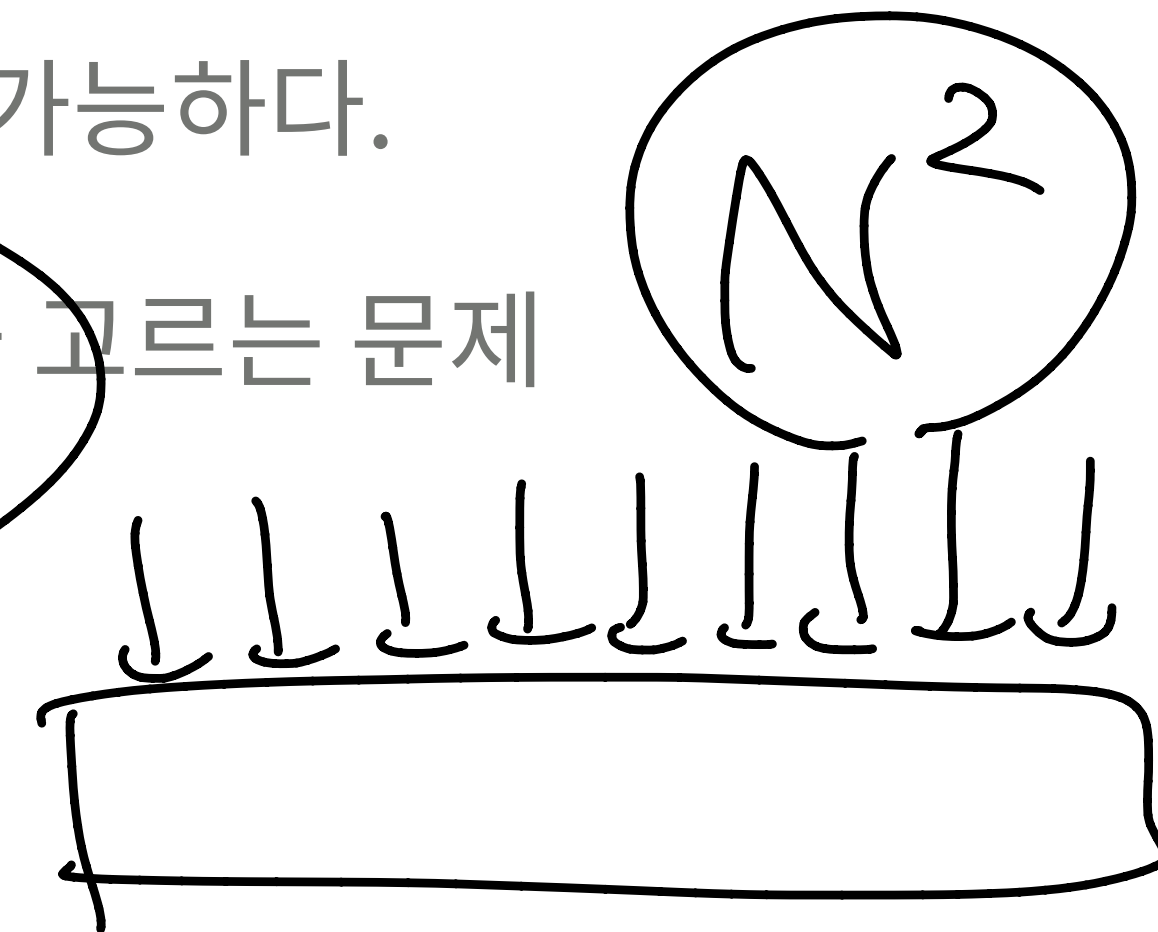
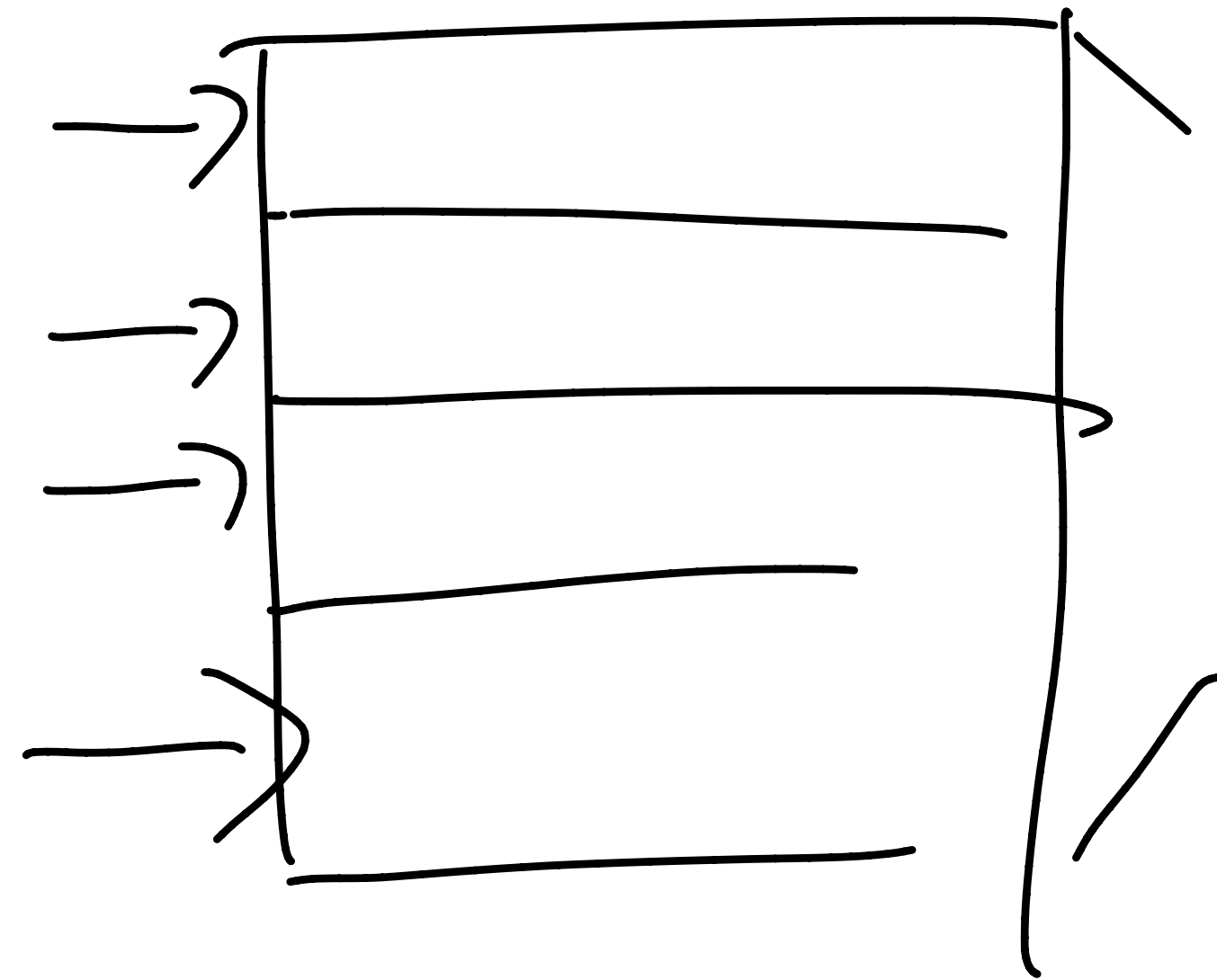
$O(4N^2)$

$\frac{O(2N^2)}{O(N^2)}$

사탕 게임

<https://www.acmicpc.net/problem/3085>

- $N \times N$ 크기의 테이블에 사탕이 있다. ($N \leq 50$)
- 인접한 두 칸을 고르고, 사탕을 교환한다. $\rightarrow (N^2) \times 2$ 가지 경우가 가능하다.
- 그 다음, 같은 색으로 이루어져 있는 가장 긴 연속 부분 행 또는 열을 고르는 문제



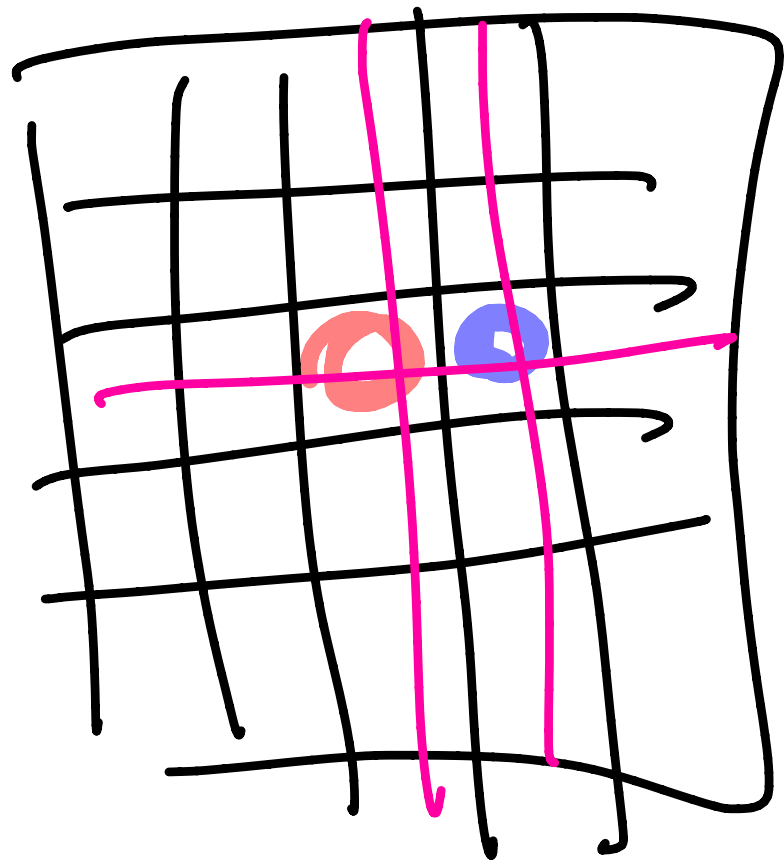
사탕 게임

26

<https://www.acmicpc.net/problem/3085>

$$50^4 = 2500^2$$

- $N \times N$ 크기의 테이블에 사탕이 있다. ($N \leq 50$)
- 인접한 두 칸을 고르고, 사탕을 교환한다. $\rightarrow (N^2) \times 2$ 가지 경우가 가능하다.
- 그 다음, 같은 색으로 이루어져 있는 가장 긴 연속 부분 행 또는 열을 고르는 문제 $\rightarrow O(N^2)$



$$\frac{N^2 \times N^2}{3N} = N^3$$

사탕 게임

<https://www.acmicpc.net/problem/3085>

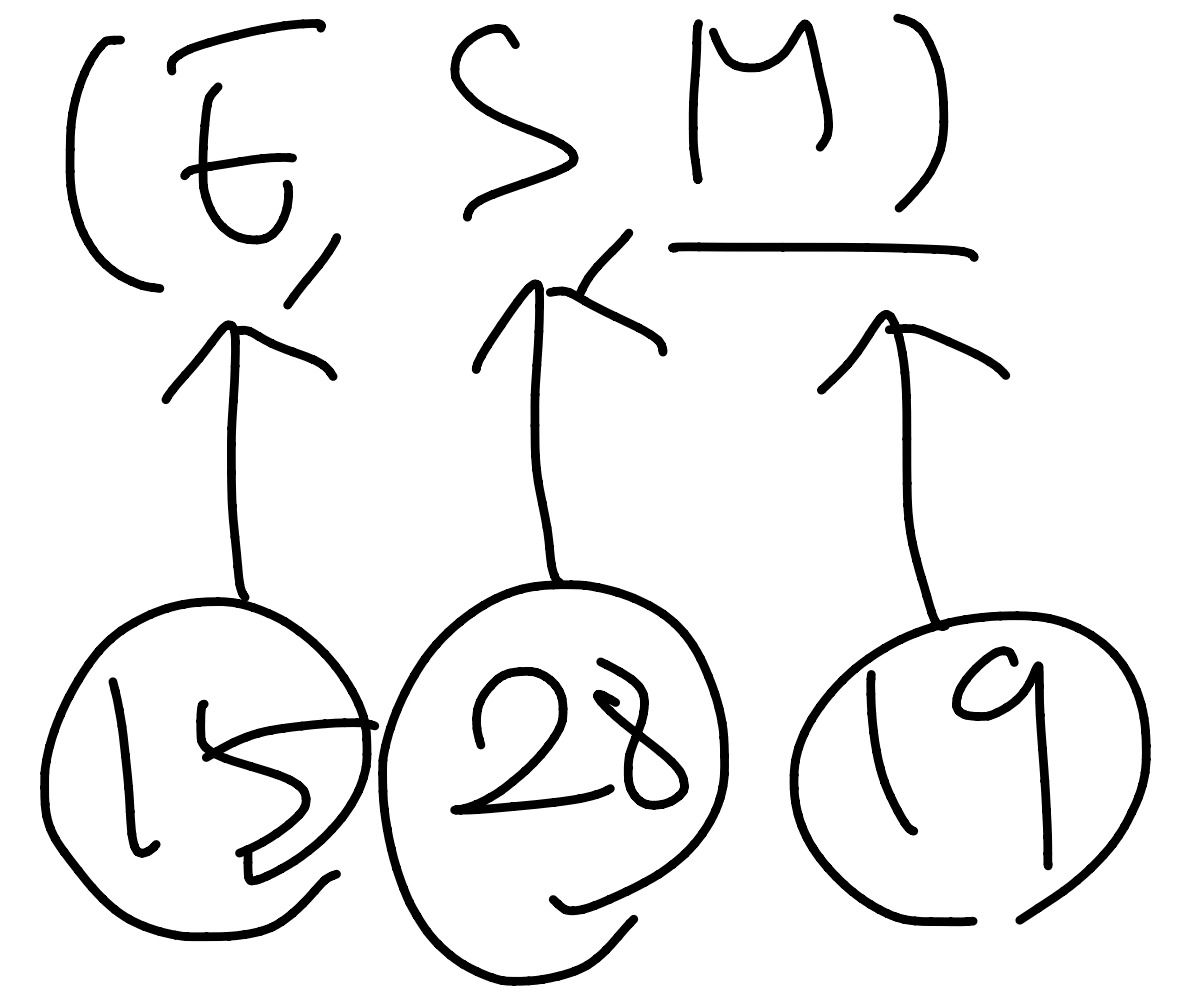
- 소스: <http://codeplus.codes/2716e00244b8425d9ff950f74fea2e7e>

날짜 계산

<https://www.acmicpc.net/problem/1476>

- 준규가 사는 나라는 E S M이라는 연도를 사용한다.
- $1 \leq E \leq 15, 1 \leq S \leq 28, 1 \leq M \leq 19$
- 1년 = 1 1 1
- 2년 = 2 2 2
- ...
- 15년 = 15 15 15
- 16년 = 1 16 16
- E S M이 주어졌을 때, 이게 몇 년인지 구하는 문제

- 17년: 2 17 17
- 18년: 3 18 18
- 19년: 4 19 19
- 20년: 5 20 1
- 21년: 6 21 2



$15 \times 28 \times 19 = 7980$

날짜 계산

<https://www.acmicpc.net/problem/1476>

- 가능한 경우의 수
- $15 \times 28 \times 19 = 7,980$
- 모든 경우를 다 해보면 된다

날짜 계산

<https://www.acmicpc.net/problem/1476>

- 소스: <http://codeplus.codes/08d18bbdfb2f4ffd957e2ed762eb4100>

날짜 계산

<https://www.acmicpc.net/problem/1476>

$$1 \leq E \leq 15$$

$$E$$

$$E \div 15$$
$$15 \div 15 = 0$$

모든 E, S, M에서 1을 빼면, 이 문제는 다음을 만족하는 가장 작은 자연수 year를 찾는 문제이다.

- $\text{year} \bmod 15 == E$
- $\text{year} \bmod 28 == S$
- $\text{year} \bmod 19 == M$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	2	3

$$(E-1) \% 15 + 1$$

$$17 \% 15 = 3$$

$$30 \% 15 = 0$$

$$30$$

$$15$$

이런식으로 year를 0부터 증가시키면서 위의 식을 검사해 구현하는 방법도 가능하다.

날짜 계산

<https://www.acmicpc.net/problem/1476>

- 소스: <http://codeplus.codes/151e08d2bcaa46a8a28f61b063550578>

날짜 계산

<https://www.acmicpc.net/problem/1476>

- 모든 E, S, M에서 1을 빼면, 이 문제는 다음을 만족하는 가장 작은 자연수 year를 찾는 문제이다.

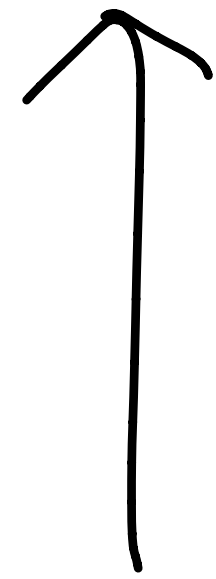
- $\text{year} \bmod 15 == E$
- $\text{year} \bmod 28 == S$
- $\text{year} \bmod 19 == M$

- 이 문제는 중국인의 나머지 정리로도 풀 수 있다.
- 중국인의 나머지 정리는 이 챕터에서 중요한 내용이 아니기 때문에, 소스 코드만 첨부한다.

날짜 계산

<https://www.acmicpc.net/problem/1476>

- 소스: <http://codeplus.codes/3208fe9c2e2f4d918f3d5d13d7794a19>



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

39

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

- 6, 7, 8이 고장났고, 5457에 이동하고 싶은 경우를 생각해보자.
- 5, 4, 3, 5, +, +, +, 5, 4, -, 7, 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 최솟값을 구하는 문제이다.
- 의미없는 것이 있는 방법은 절대 최소가 될 수 없다.

8455++

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

41

<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

42

<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

43

<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

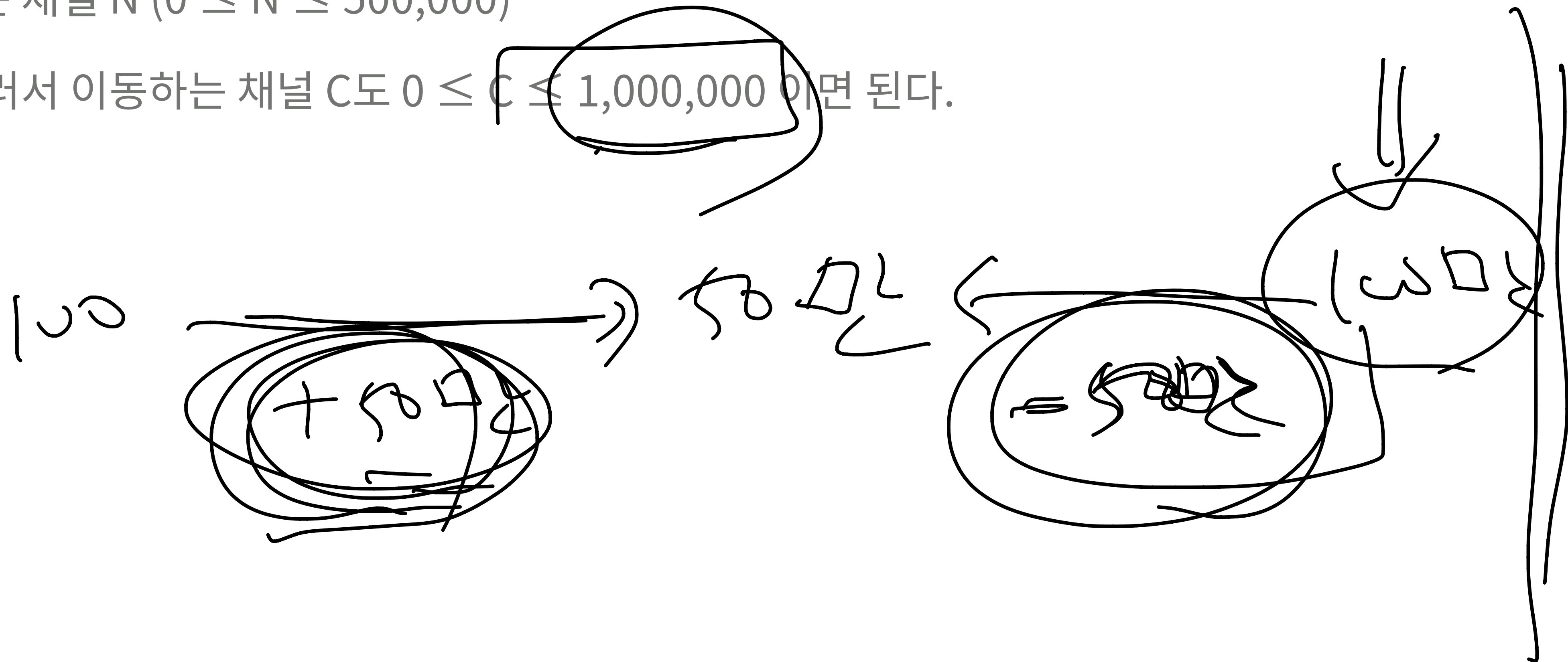
<https://www.acmicpc.net/problem/1107>

- 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 C도 $0 \leq C \leq 1,000,000$ 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

50

<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

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

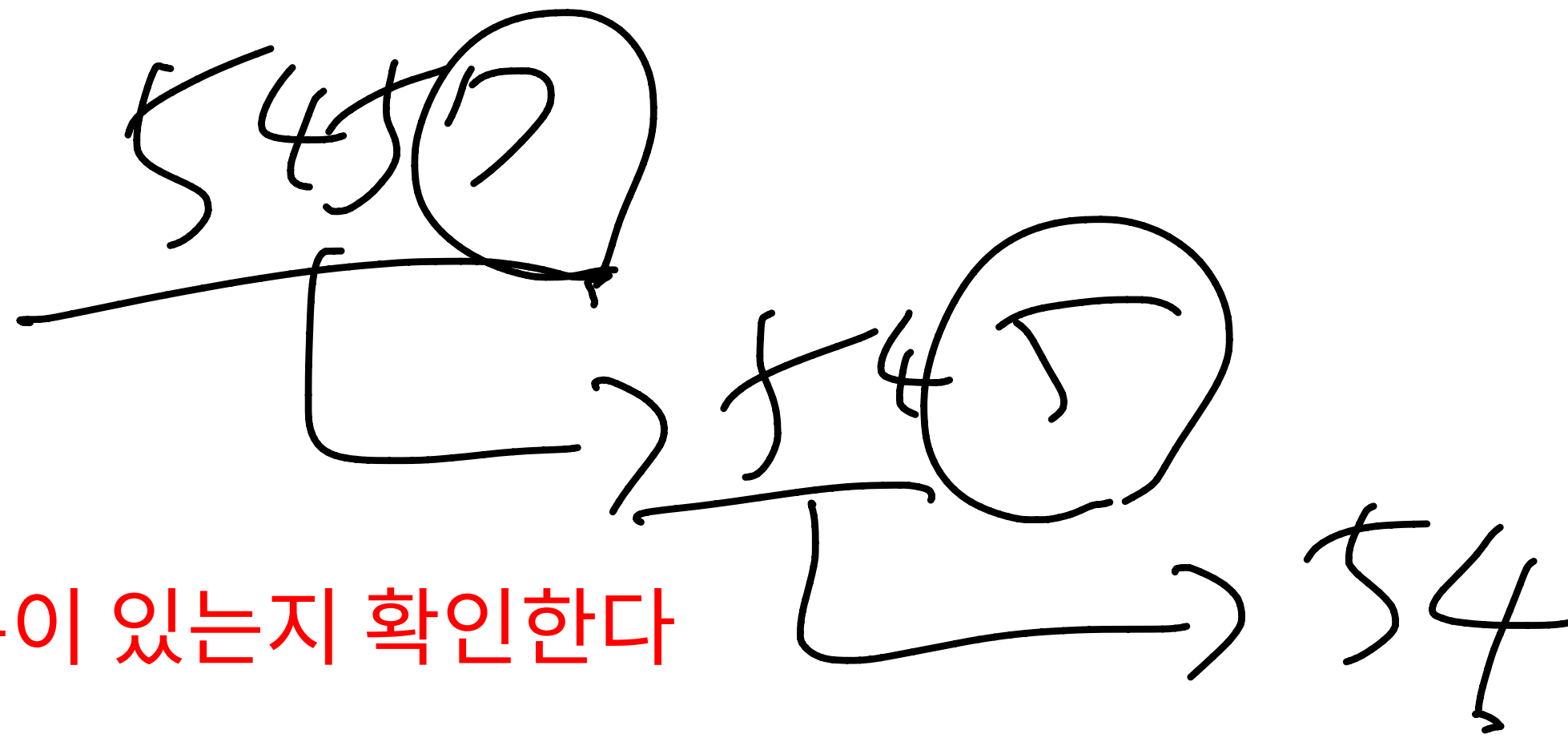
<https://www.acmicpc.net/problem/1107>

1. 이동할 채널 C 를 정한다
2. C 에 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

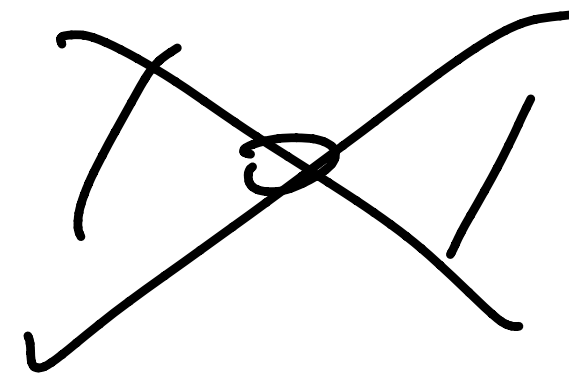
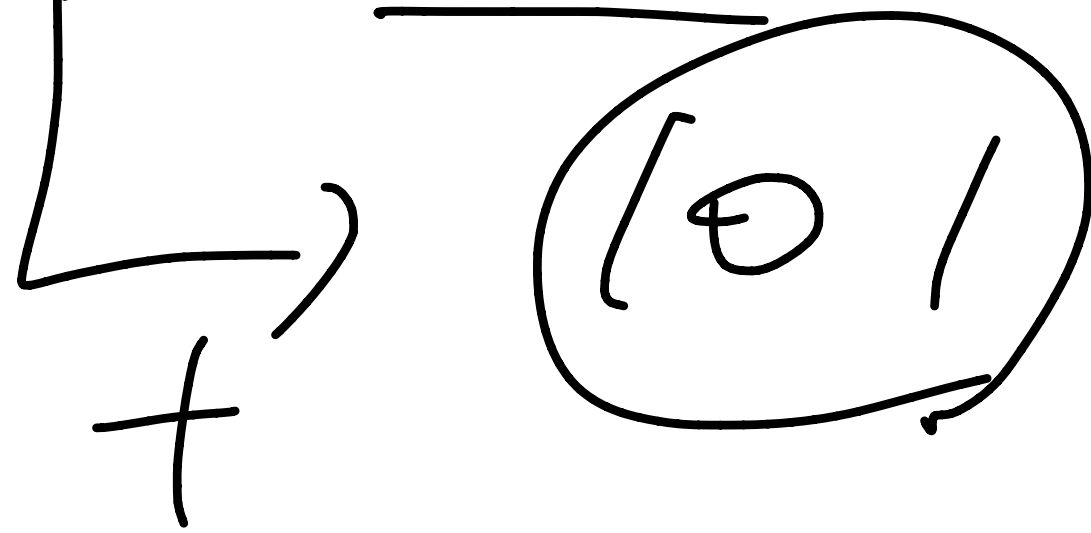
```
int possible(int c) {  
    if (c == 0) {  
        return broken[0] ? 0 : 1;  
    }  
    int len = 0;  
    while (c > 0) {  
        if (broken[c % 10]) return 0;  
        len += 1;  
        c /= 10;  
    }  
    return len;  
}
```

N-C

리모컨

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

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



리모컨

58

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

- 소스: <http://codeplus.codes/ed3372d551b749759cc79ae6dbf7845f>

테트로미노

<https://www.acmicpc.net/problem/14500>

59

$$\frac{(9+2)}{19 \times 500^2}$$

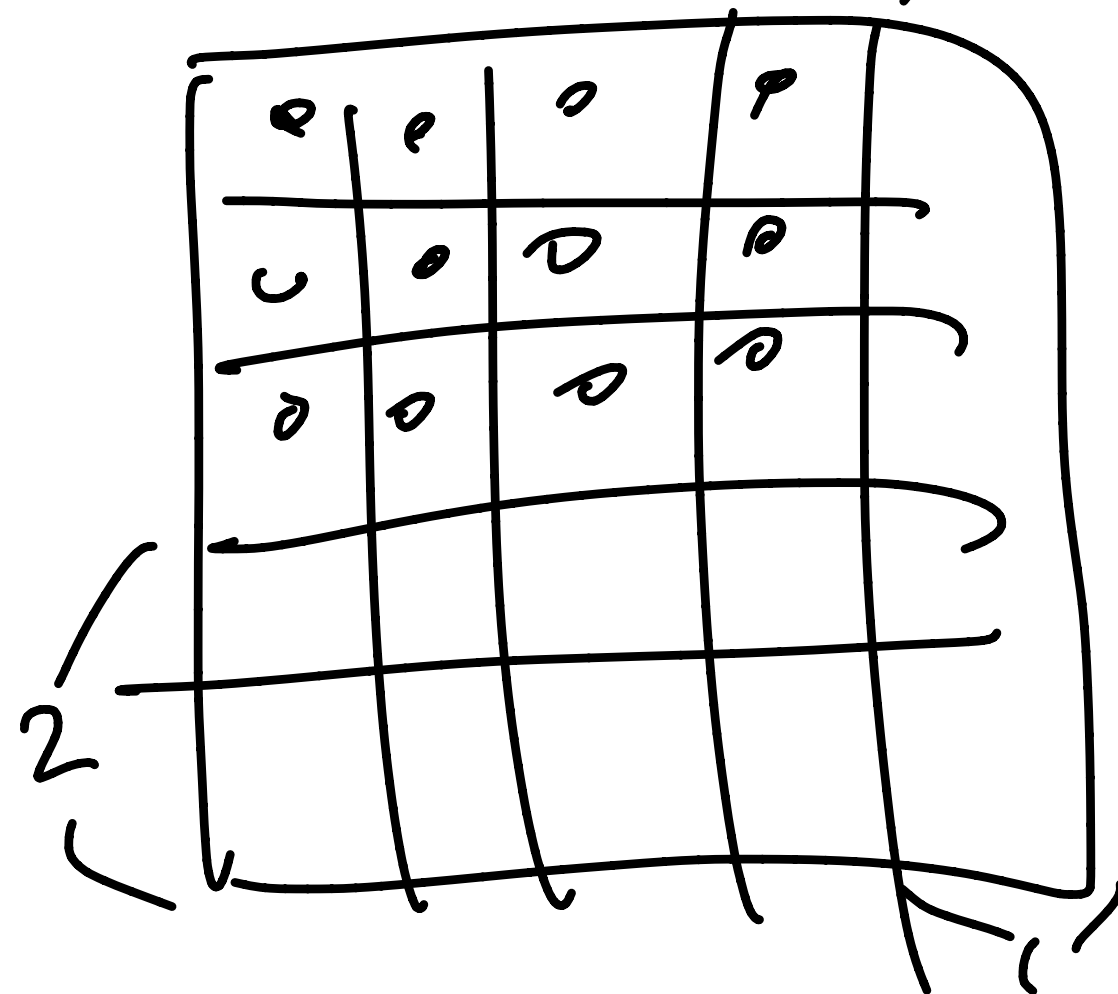
• 폴리오미노는 크기가 1×1 인 정사각형을 여러 개 이어 붙여서 만든 도형이다.

• 정사각형 4개를 이어 붙인 폴리오미노는 테트로미노라고 하며, 총 5가지가 있다. 2

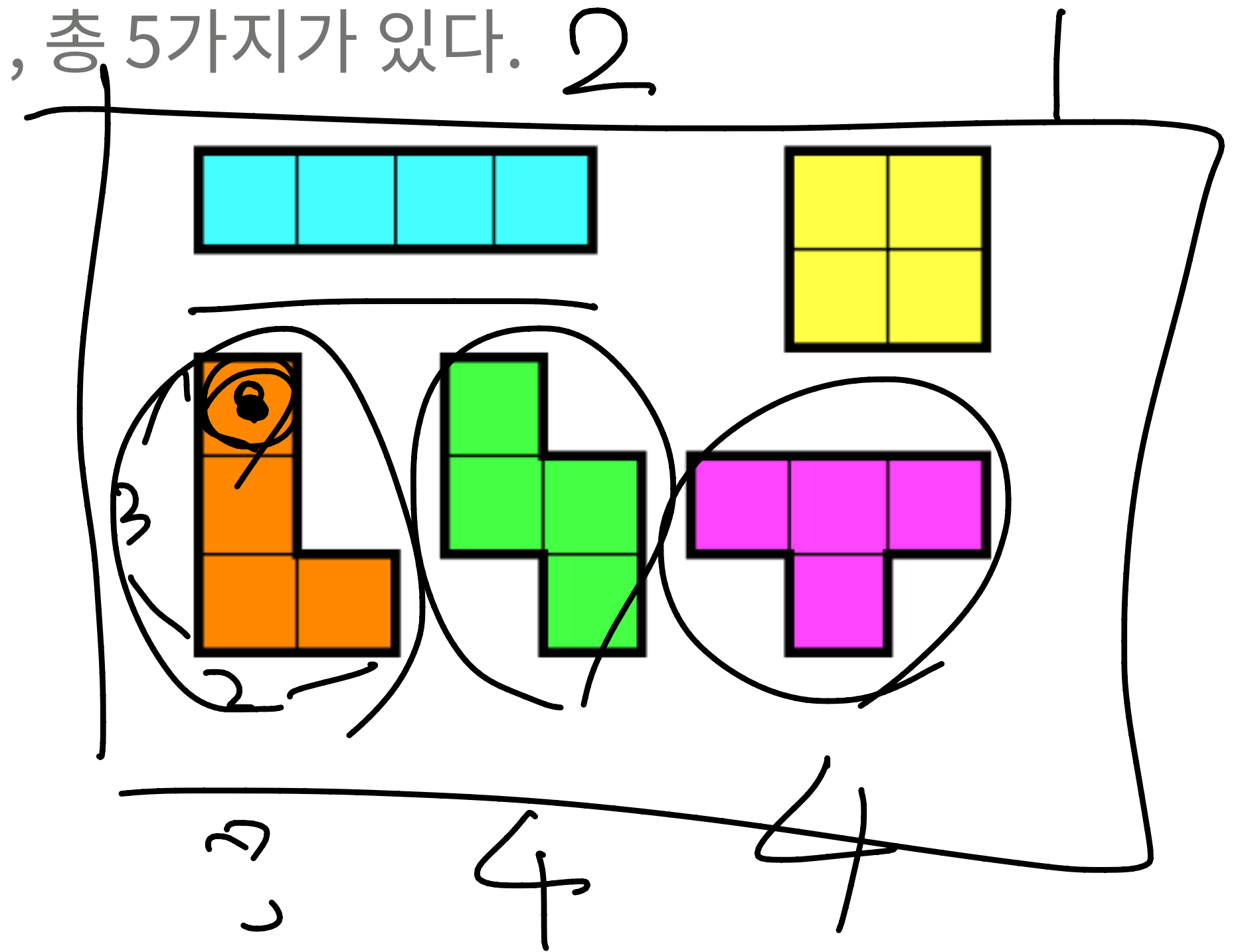
• $N \times M$ 크기의 종이 위에 테트로미노를 하나 놓아서

• 놓인 칸에 쓰여 있는 수의 합을 최대로 하는 문제

• $4 \leq N, M \leq 500$



$$\frac{O(NM)}{19 \times N \times M}$$



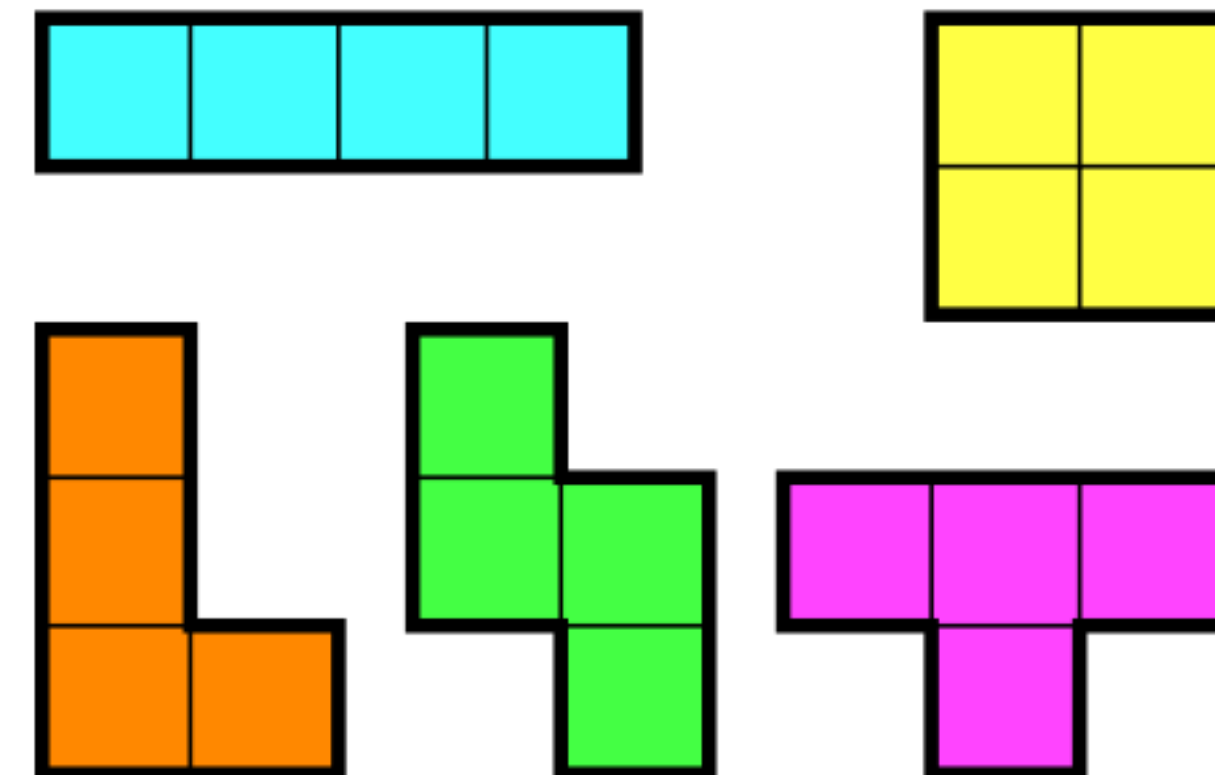
테트로미노

<https://www.acmicpc.net/problem/14500>

60

- 테트로미노는 총 19가지가 있고
- 하나의 테트로미노당 놓을 수 있는 방법의 개수는 약, $O(NM)$ 가지 이다
- 경우의 수가 많지 않기 때문에
- 각각의 테트로미노에 대해서 모든 칸에 놓아본다

475 0000



$$\begin{array}{r} 19 \times 500^2 \\ 20 \times 500 \times 500 \\ \hline 10000 \end{array}$$

500 0000

테트로미노

<https://www.acmicpc.net/problem/14500>

- 소스: <http://codeplus.codes/79d5e637320f44a7a99315854aa7e345>
 - 소스 2: <http://codeplus.codes/cf6b35eb8b384f9289fe70049545cdef>
 - 소스 3: <http://codeplus.codes/503fa1dac4d14d3ab1b554f49c473120>
-
- 소스 3과 같이 구현할 수도 있지만, 이 문제에서는 적절하지 않다.

건너 뛰며 해보기

카잉 달력

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

63

• 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, y 가 주어졌을 때, $\langle x:y \rangle$ 이 몇 번째 해인지 구하는 문제

$$M < N$$

PSW

$$\dots \langle M, M \rangle \rightarrow \langle 1, M+1 \rangle$$

$$\underline{M}, \underline{N} \leq 4000$$

$$16^3 = 16^3$$

카잉 달력

64

<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

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

65

- $M = 5$
- $N = 7$

$\langle \textcircled{3} 6 \rangle$

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

카잉 달력

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

66

$$\left(\tau \div M, \tau \div N \right)$$
$$\left(\underline{32 \div 5}, \underline{32 \div 7} \right)$$

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

카잉 달력

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

$O(N)$

67

- $M = 5$
- $N = 7$
- $x = 3$
- $y = 2$

- | | | | |
|------------|-------------|-------------|-------------|
| • 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>

- x를 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/f501ae2d195d418bbe75b4ab6c36e710>

수 이어 쓰기 1

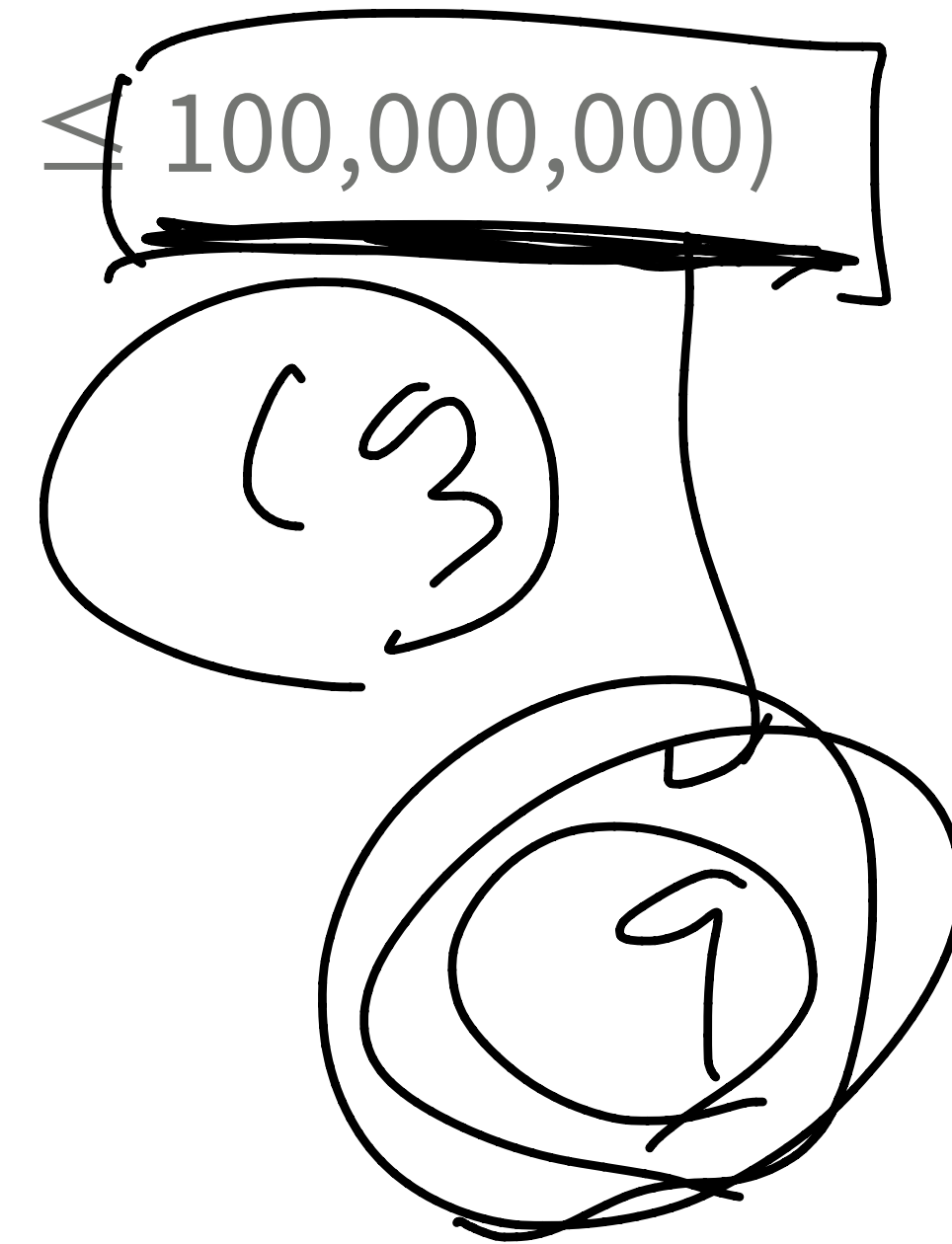
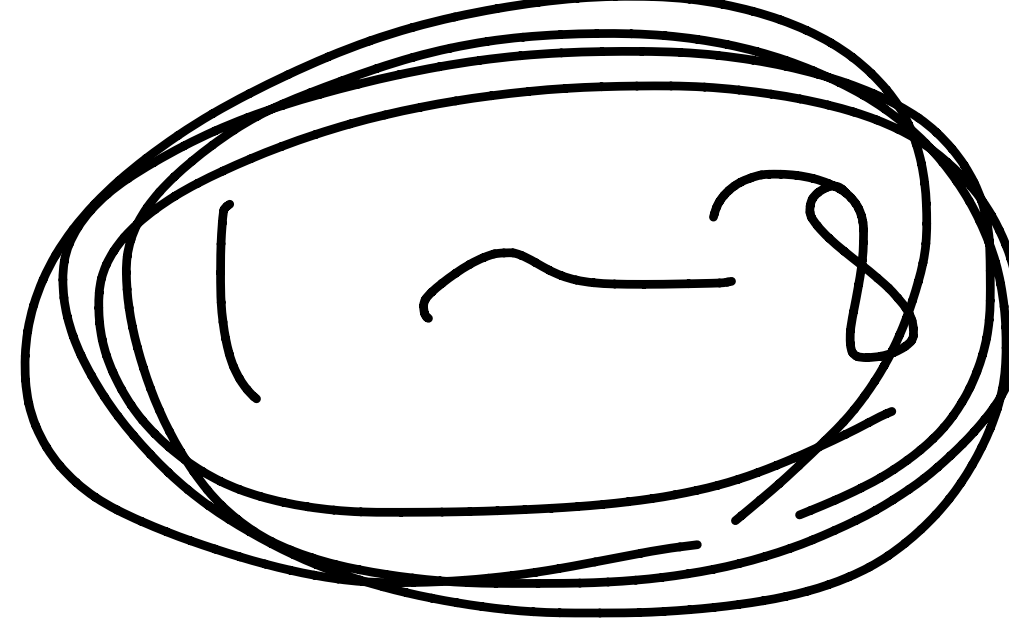
<https://www.acmicpc.net/problem/1748>

70

- 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?



2

수 이어 쓰기 1

71

<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.
- $O(N) \times 10$
- 여기서 10은 수의 최대 자릿수를 의미한다.

수 이어 쓰기 1

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

- 수의 자리수별로 나누어서 문제를 해결할 수 있다.
- $N = 120$

$$\begin{array}{l}
 \bullet \text{ 1 - 9} \quad 1 \times 9 \\
 \bullet \text{ 10 - 99} \quad 2 \times (99 - 10 + 1) \\
 \bullet \text{ 100 - 120} \quad 3 \times (120 - 100 + 1)
 \end{array}$$

수 이어 쓰기 1

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

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

수 이어 쓰기 1

75

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

- 소스: <http://codeplus.codes/5d4c53a6cde44bb3ac01c65d13651cb5>

N중 for문

N중 for문

for

- N개 중에 일부를 선택해야 하는 경우에 많이 사용한다
- 재귀 호출이나 비트마스크를 사용하면 더 간결하고 보기 쉬운 코드를 작성할 수 있기 때문에, 사용할 일이 거의 없다.

1, 2, 3 더하기

<https://www.acmicpc.net/problem/9095>

78

$$n \leq 10$$

- 정수 n 을 1, 2, 3의 합으로 나타내는 방법의 수를 구하는 문제

- $n = 4$

- $1+1+1+1$

- $1+1+2$

- $1+2+1$

- $2+1+1$

- $2+2$

- $1+3$

- $3+1$

$$\begin{array}{c} \textcircled{1,2,3} \\ \uparrow \end{array} + \textcircled{1,2,3} + \dots + \textcircled{1,2,3}$$

1, 2, 3 더하기

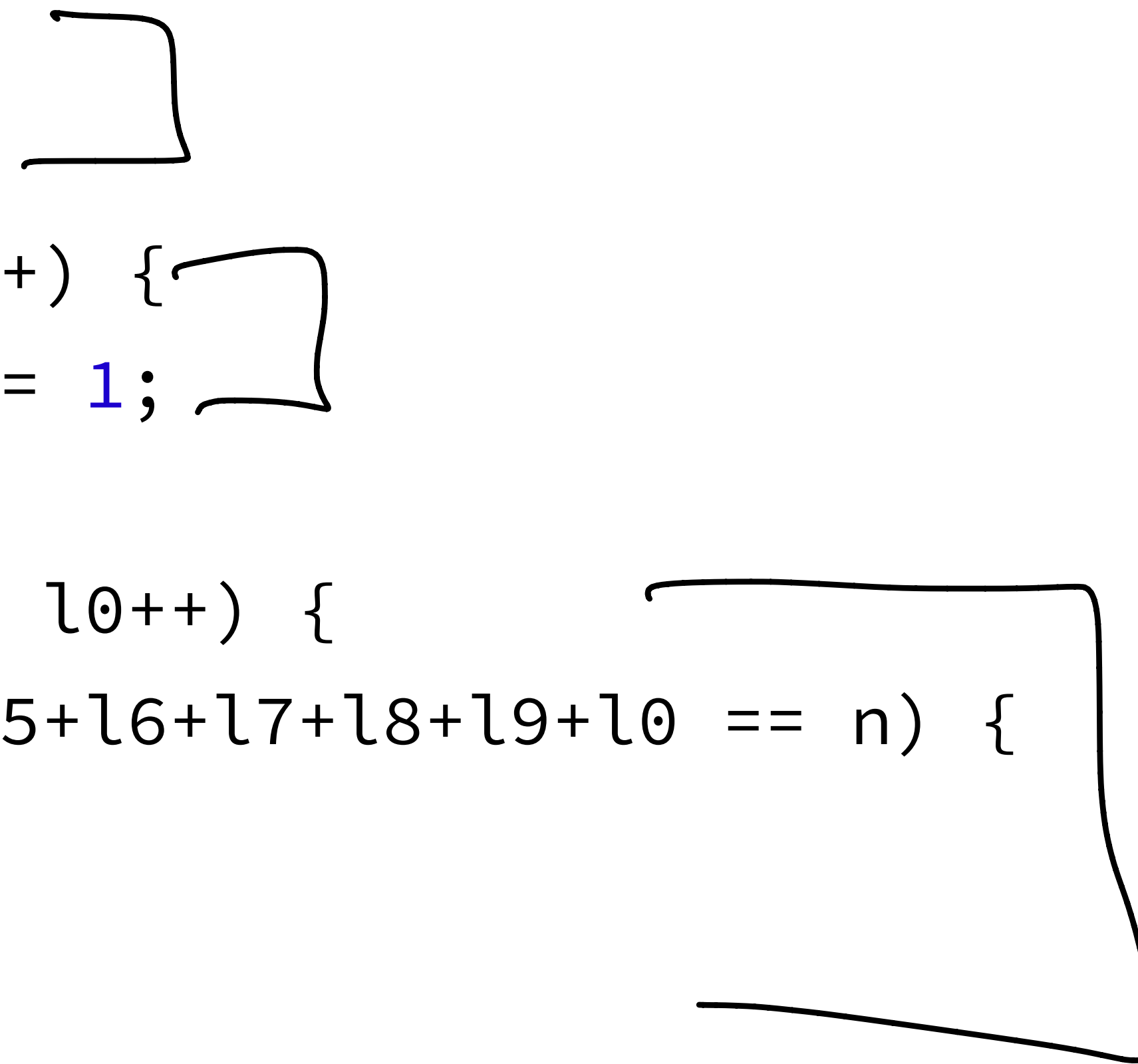
<https://www.acmicpc.net/problem/9095>

- N이 10보다 작거나 같기 때문에
- 최대 10개 이하로 표현 가능
- $1+1+1+1+1+1+1+1+1+1$
- 10중 for문!

1, 2, 3 더하기

<https://www.acmicpc.net/problem/9095>

```
for (int l1=1; l1<=3; l1++) {  
    if (l1 == n) ans += 1;  
    for (int l2=1; l2<=3; l2++) {  
        if (l1+l2 == n) ans += 1;  
        ... 생략  
        for (int l0=1; l0<=3; l0++) {  
            if (l1+l2+l3+l4+l5+l6+l7+l8+l9+l0 == n) {  
                ans += 1;  
            }  
        }  
    }  
}
```



1, 2, 3 더하기

<https://www.acmicpc.net/problem/9095>

- 소스: <http://codeplus.codes/16c62ce433a74909ace1620fa0014da2>

끝

코드 플러스

<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.