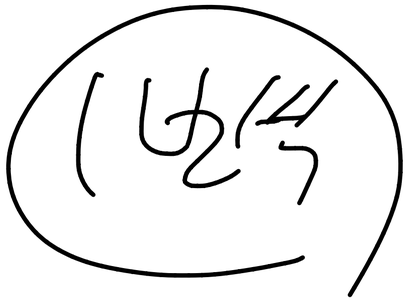
8/19 - 1

최백준 choi@startlink.io

브루트 포스

브루트 포스

Brute Force



- 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

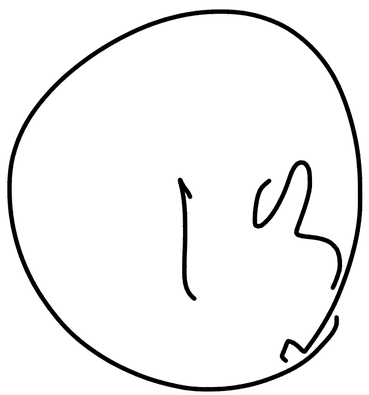
Brute Force

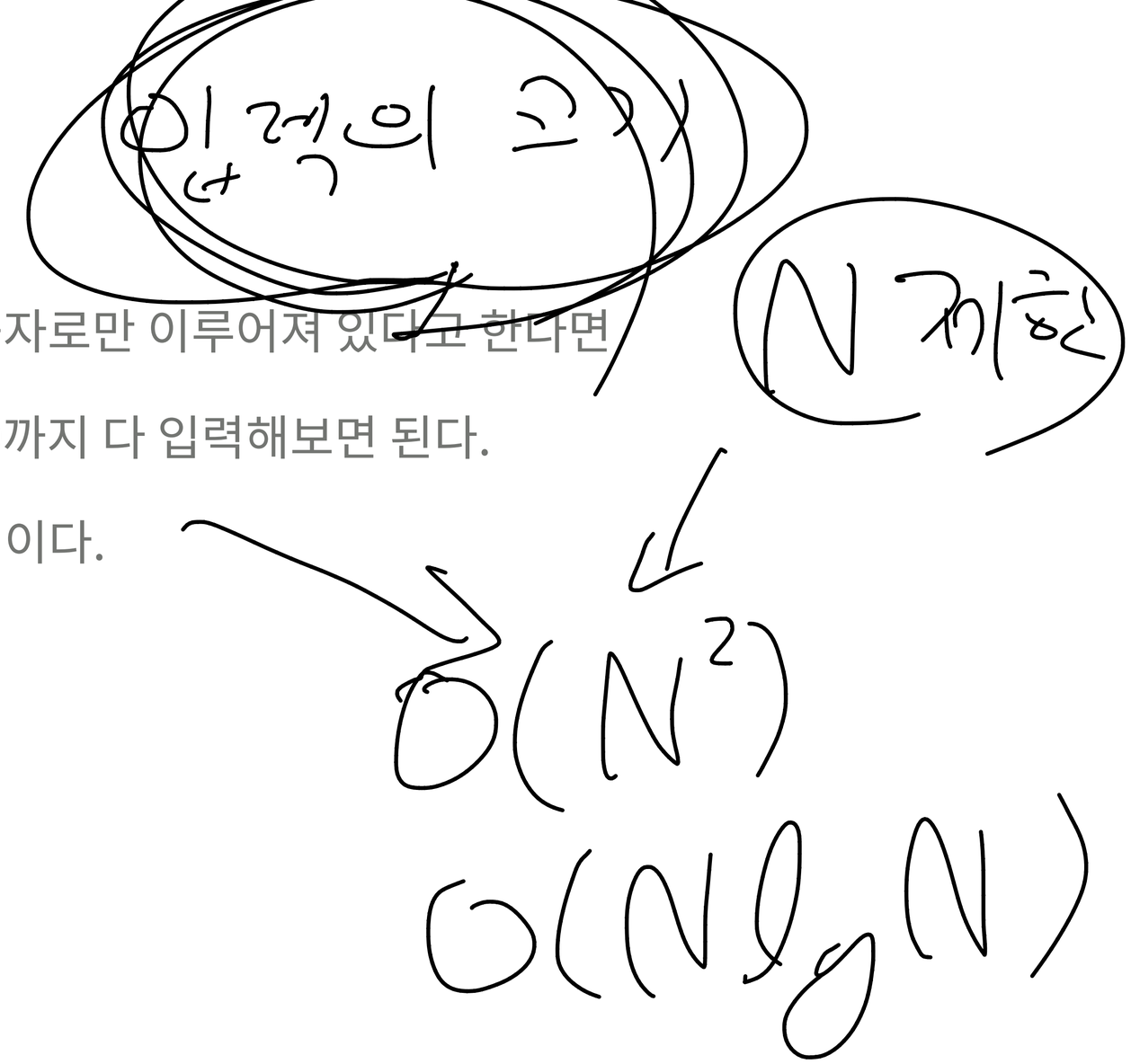
- 예를 들어, 비밀번호가 4자리이고, 숫자로만 이루어져 있다고 한다면
- 0000부터 9999까지 다 입력해보면 된다.
- 경우의 수가 10,000가지 이다.
- 사람이 직접 비밀번호를 입력하는데 1초가 걸린다면 10,000초 = 2.7시간 정도 걸린다.

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.





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.
- 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

- 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(경우의 수 * 방법 1개를 시도해보는데 걸리는 시간 복잡도)가 걸린다.

경우의 수

경우의 수

Brute Force

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

경우의 수

Brute Force

- N명의 사람이 한 줄로 서는 경우의 수 → $N!$
- N명의 사람 중에서 대표 두 명을 뽑는 경우의 수 → $_NC_2 = \frac{N(N-1)}{2} = N^2$
- N명의 사람 중에서 대표 세 명을 뽑는 경우의 수 → N^3
- N명의 사람 중에서 반장 1명과 부반장 1명을 뽑는 경우의 수 → N^2
- N명의 사람이 있을 때, 각 사람이 영화를 볼지, 보지 않을지 결정한다. 가능한 조합의 수 → 2^N

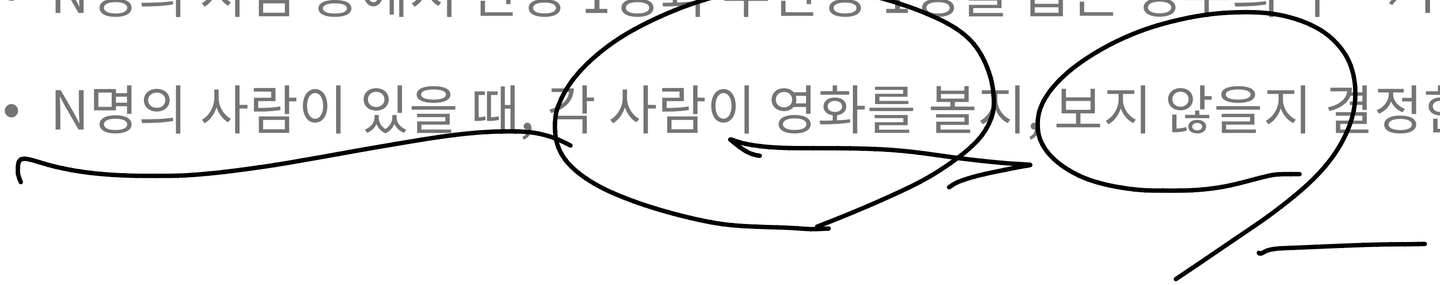
$N \le 500$

$N \le 20$

경우의 수

Brute Force

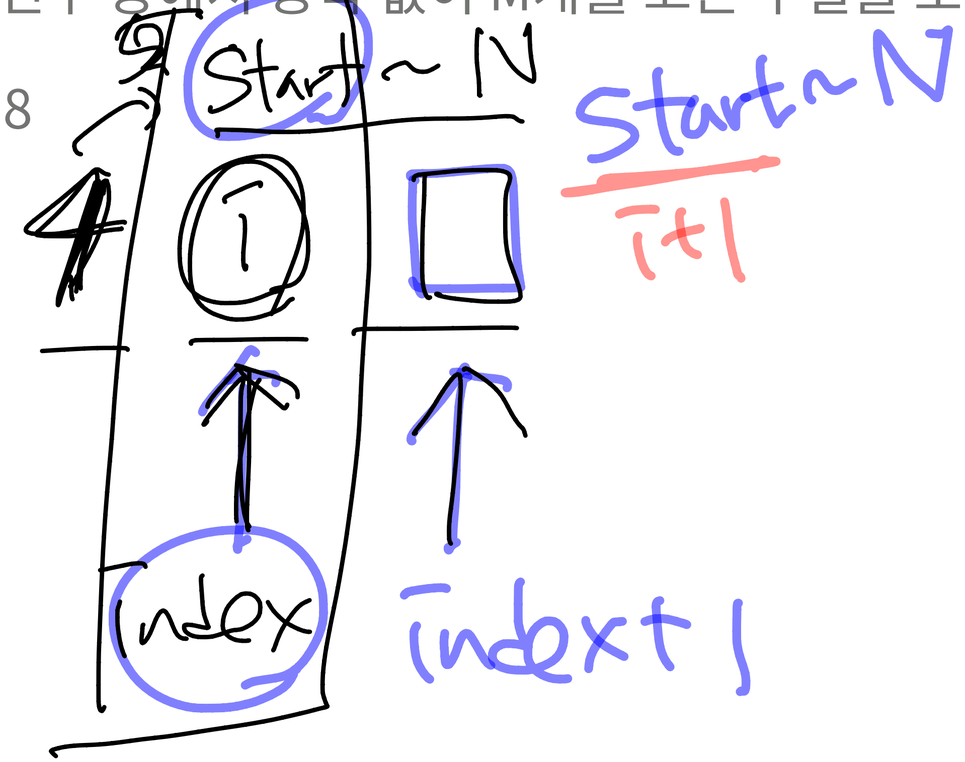
- N명의 사람이 한 줄로 서는 경우의 수 → $N \times (N-1) \times \cdots \times 1 = N!$
- N명의 사람 중에서 대표 두 명을 뽑는 경우의 수 → $N \times (N-1) / 2$
- N명의 사람 중에서 대표 세 명을 뽑는 경우의 수 → $N \times (N-1) \times (N-2) / 3!$
- N명의 사람 중에서 반장 1명과 부반장 1명을 뽑는 경우의 수 → $N \times (N-1)$
- N명의 사람이 있을 때, 각 사람이 영화를 볼지, 보지 않을지 결정한다. 가능한 조합의 수 →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

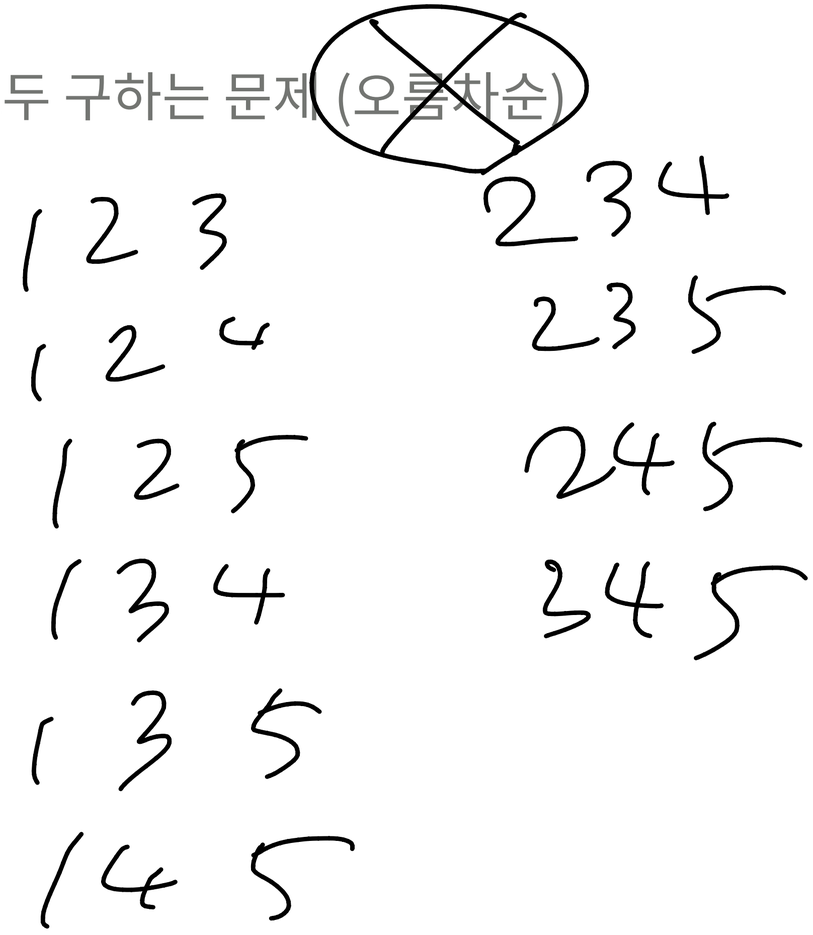


N과 M (2)

https://www.acmicpc.net/problem/15650

- 1부터 N까지 자연수 중에서 중복 없이 M개를 고른 수열을 모두 구하는 문제 (오름차순)
- $1 \le M \le N \le 8$

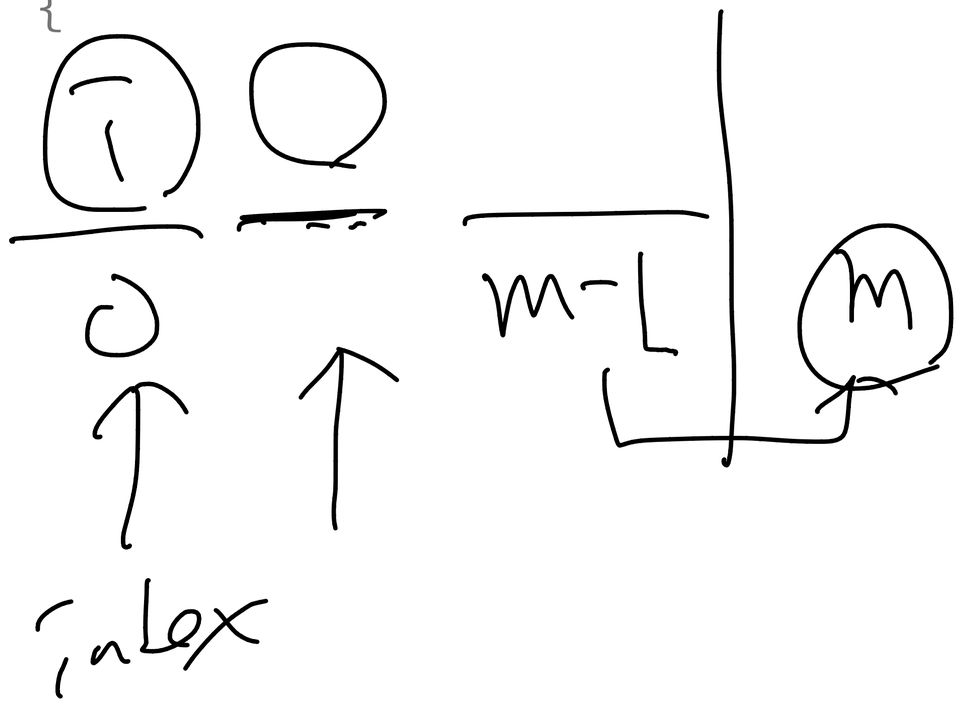




N과 M (2)

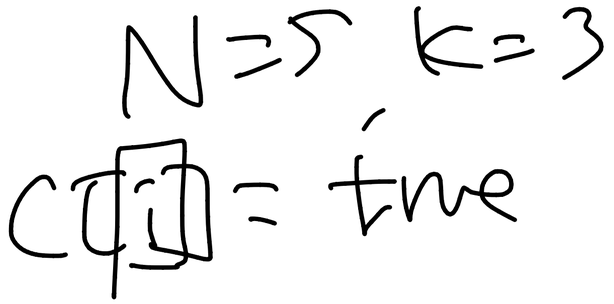
https://www.acmicpc.net/problem/15650

```
bool c[10]; int a[10];
void go(int index, int start, int n, int m) {
    if (index == m) {
        // 수열을 출력
        return;
    }
    for (int i=start; i<=n; i++) {
        if (c[i]) continue;
        c[i] = true; a[index] = i;
        go(index+1, i+1, n, m);
        c[i] = false;
    }
}
// go(0, 1, n, m);
```

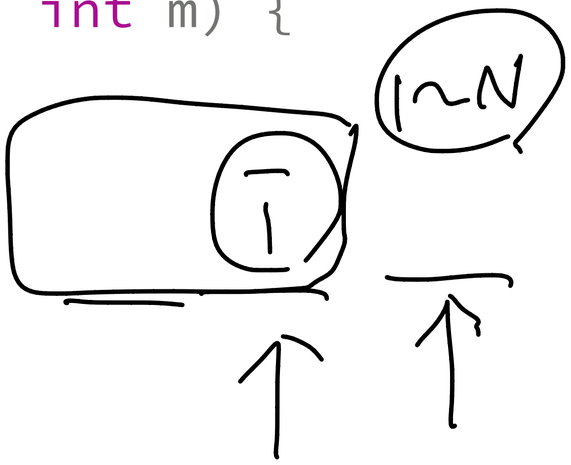


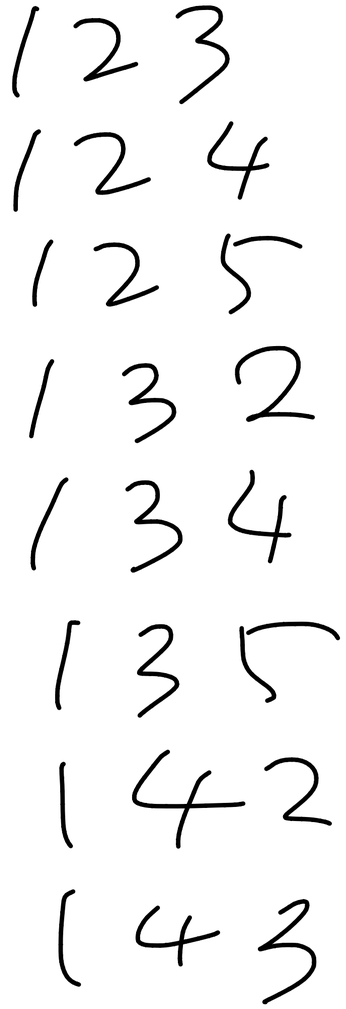
N과 M (2)

https://www.acmicpc.net/problem/15650




```
bool c[10]; int a[10];
void go(int index, int start, int n, int m) {
    if (index == m) {
        // 수열을 출력
        return;
    }
    for (int i=start; i<=n; i++) {
        if (c[i]) continue;
        c[i] = true; a[index] = i;
        go(index+1, start, n, m);
        c[i] = false;
    }
}
// go(0, 1, n, m);
```

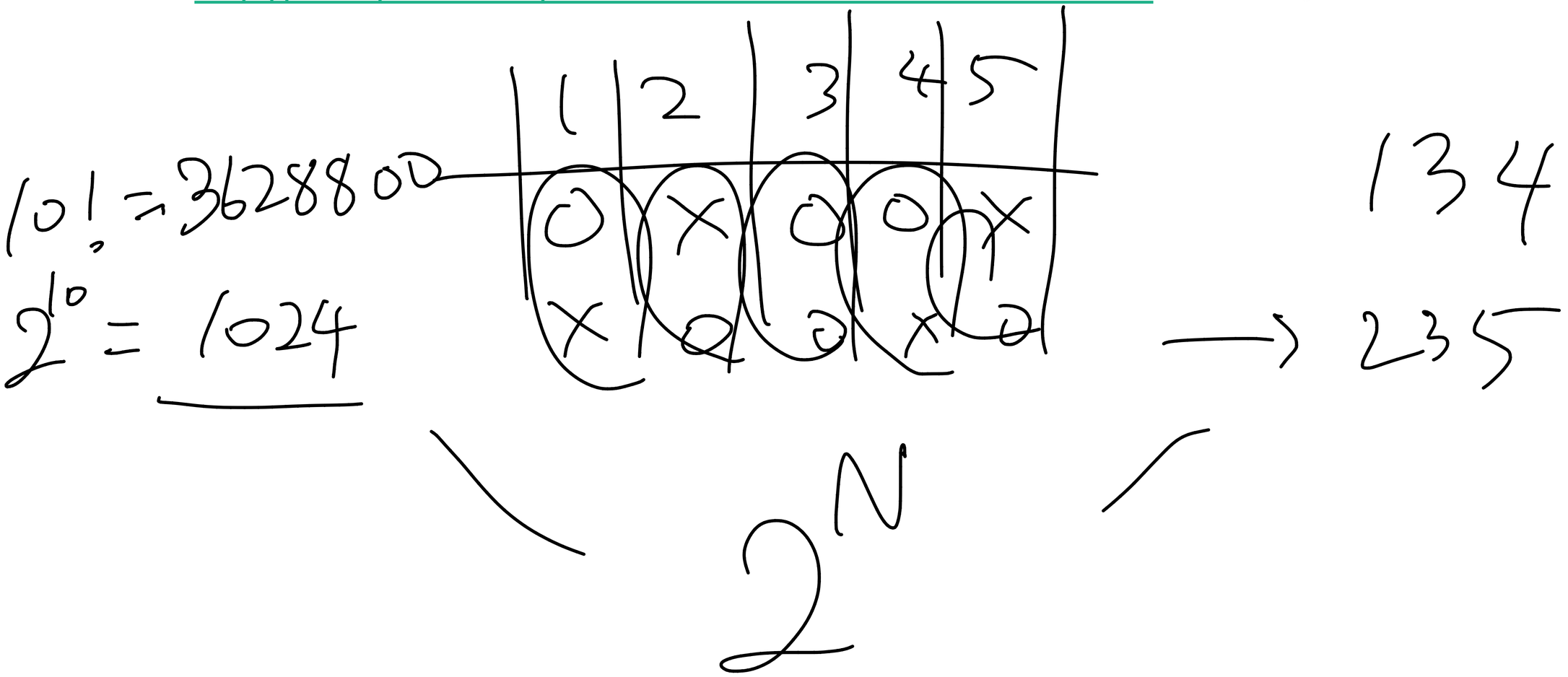




N과 M (2)

https://www.acmicpc.net/problem/15650

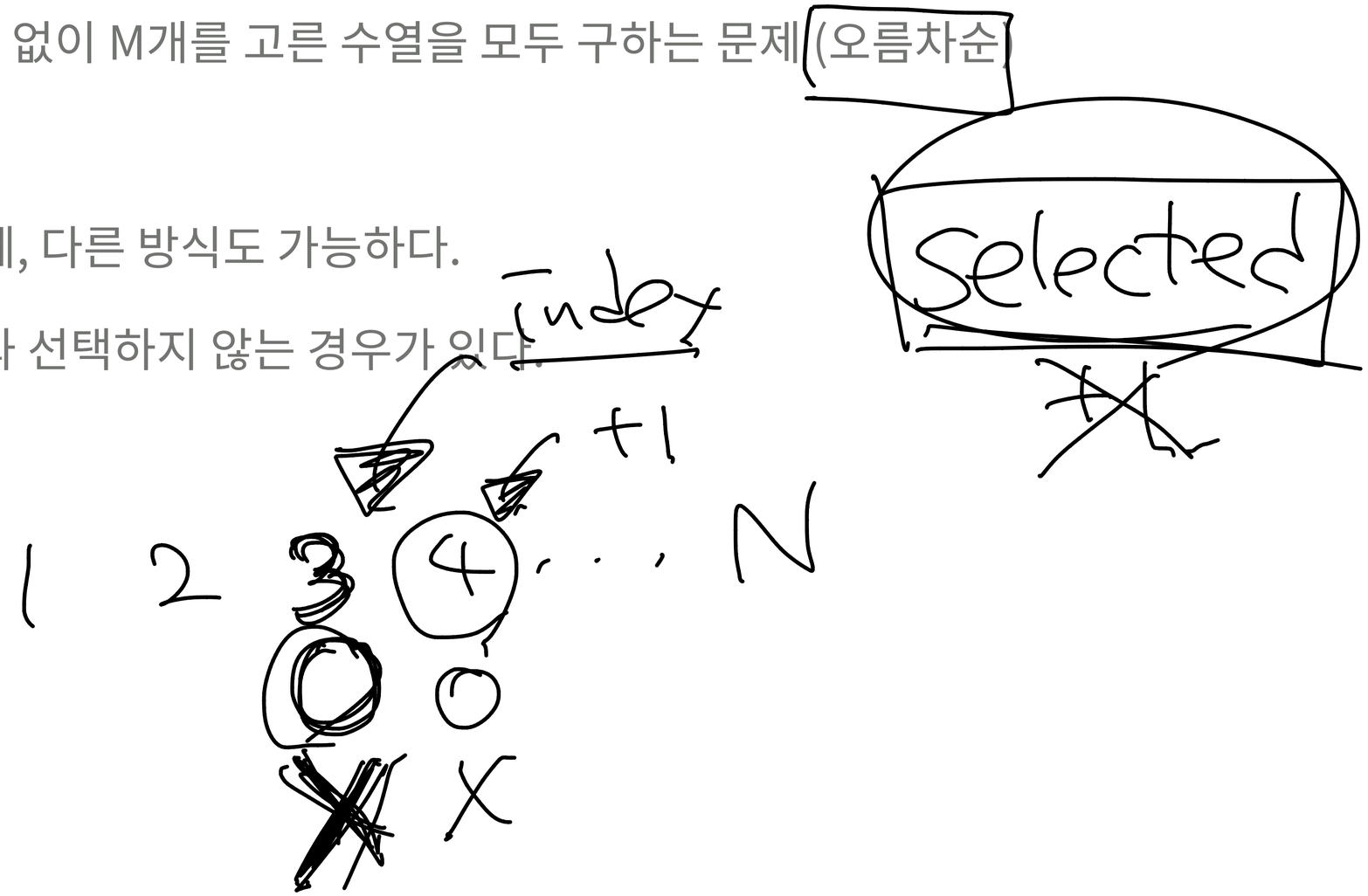
- 소스: http://codeplus.codes/987fdf7e666f490a860e5b755cc1b885



N과 M (2)

https://www.acmicpc.net/problem/15650

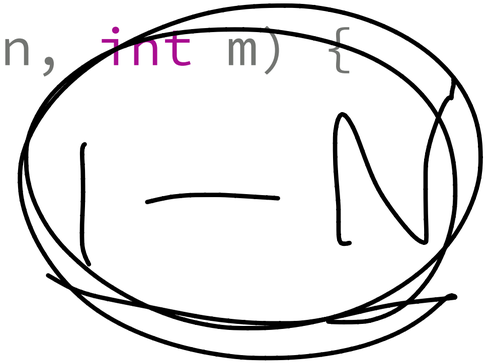
- 1부터 N까지 자연수 중에서 중복 없이 M개를 고른 수열을 모두 구하는 문제 (오름차순)
- $1 \le M \le N \le 8$
- 오름차순만 고르는 것이기 때문에, 다른 방식도 가능하다.
- 각각의 자연수를 선택하는 경우와 선택하지 않는 경우가 있다.



N과 M (2)

https://www.acmicpc.net/problem/15650

```
int a[10];
void go(int index, int selected, int n, int m) {
    if (selected == m) {
        // 수열 출력
        return;
    }
    if (index > n) return;
    a[selected] = index;
    go(index+1, selected+1, n, m);
    a[selected] = 0;
    go(index+1, selected, n, m);
}
// go(1, 0, n, m);
```



N과 M (2)

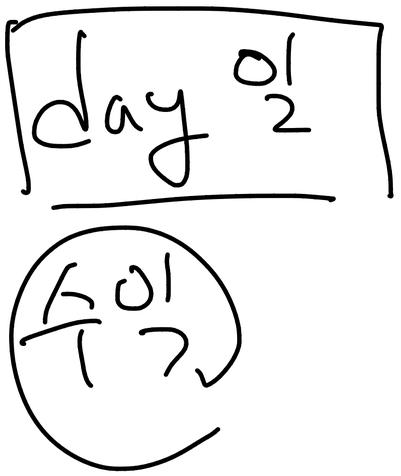
https://www.acmicpc.net/problem/15650

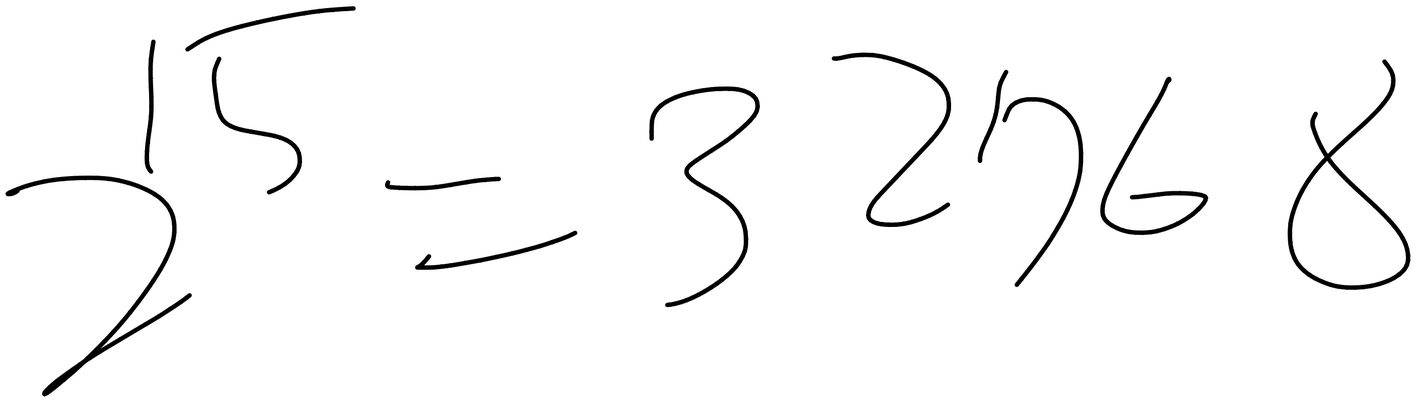
- 소스: http://codeplus.codes/e9ea6b4227da44c382f2abc3ae4ae1fb

퇴사

https://www.acmicpc.net/problem/14501

- N+1일이 되는 날 퇴사를 하려고 한다 ($1 \le N \le 15$)
- 남은 N일 동안 최대한 많은 상담을 하려고 한다
- 하루에 하나의 상담을 할 수 있고
- i일에 상담을 하면, T[i]일이 걸리고 P[i]원을 번다





퇴사

https://www.acmicpc.net/problem/14501

- go(day, sum)
  - day일이 되었다. day일에 있는 상담을 할지 말지 결정해야 한다
  - 지금까지 얻은 수익은 sum이다

① 정답을 찾은 경우

day == n+1

② 불가능한 경우

day > n+1

③ 다음 경우 호출

1. 상담을 함 : go(day + T[day], sum + P[day])
2. 상담을 안함 : go(day+1, sum)

퇴사

https://www.acmicpc.net/problem/14501

- go(day, sum)
  - day일이 되었다. day일에 있는 상담을 할지 말지 결정해야 한다.
  - 지금까지 얻은 수익은 sum이다
- 정답을 찾은 경우
  - day == n
- 불가능한 경우
  - day > n
- 다음 경우
  - 상담을 한다: go(day+t[day], sum+p[day])
  - 상담을 하지 않는다: go(day+1, sum)

퇴사

https://www.acmicpc.net/problem/14501

- 소스: http://codeplus.codes/8a59441ae65f43c7ba91ca2ef2c84fd5