

01. 완전탐색(기본)

완전 탐색이란?

모든 경우의 수를 다 해보는 방법

영어로 **Brute-Force**(무식하게 풀기)라고도 한다.

<종류>

-순열

-백트래킹

-BFS

완전 탐색이란?

오잉? 이것도 알고리즘이에요?

-> 컴퓨터의 빠른 계산 능력을 이용하는 알고리즘

4자리 암호를 사람이 직접 입력할 때

-> 0000 ~ 9999



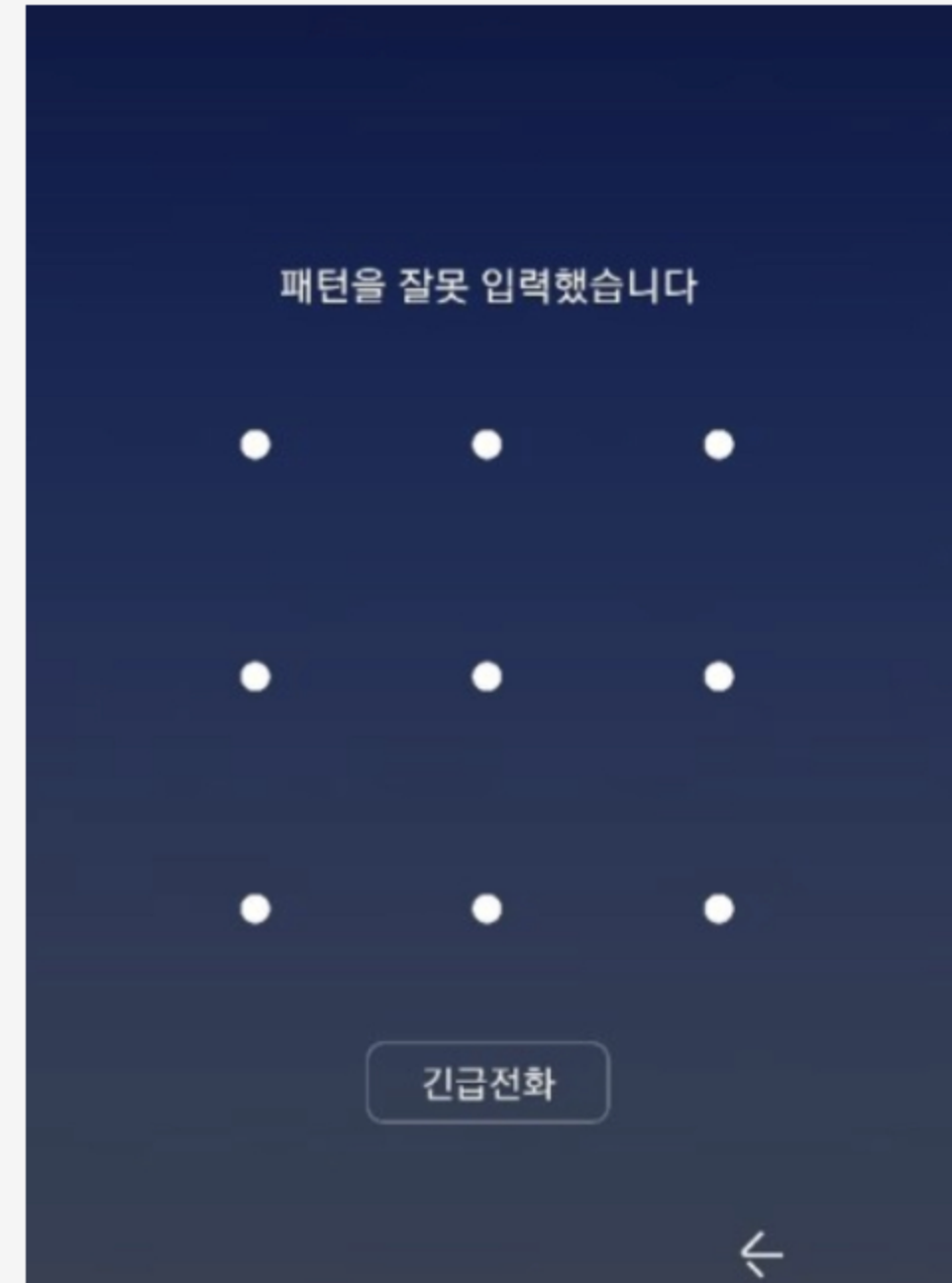
완전 탐색이란?

오잉? 이것도 알고리즘이에요?

-> 컴퓨터의 빠른 계산 능력을 이용하는 알고리즘

9개의 점을 연결하는 패턴

-> $2^9 \times 9! =$ 약 1억 8천 가짓수



완전 탐색 구현 과정

1. 가능한 모든 가짓수를 계산해본다.

입력, 출력 제한이 중요!

2. 어떤 식으로 구현을 할 지 생각한다.

단순 for문 사용, 순열, 재귀(백트래킹) 등이 있다.

3. 답을 구하고, 제출해 본다.

시간 계산

완전 탐색 방법은 무식한 방법으로 사람은 편하지만 기계는 힘들다!

-> 적용하기 전에 시간은 우리가 계산해주어야 한다!

보통 1억번의 연산 = 1초

$O(N)$: 1억

$O(N \log N)$: 5백만

$O(N^2)$: 1만

$O(N!)$: 11

시간 제한

1 초

기본 코드 1. 단순 for문

P. 포커 카드 N장 중 2장을 골라 두 장의 합이 M에 최대한 가까운 합을 구하시오!

N장 중 2장 고르기 $\rightarrow nC_2$

```
int n, m; cin >> n >> m;
int sum, ans = 0;
for (int i = 0; i < n; i++) {
    for (int j = i + 1; j < n; j++) {
        sum = arr[i] + arr[j];
        if (abs(m - ans) > abs(m - sum)) {
            ans = sum;
        }
    }
}
```

기본 코드 1. 단순 for문

P. 포커 카드 N장 중 2장을 골라 두 장의 합이 M에 최대한 가까운 합을 구하시오!

N장 중 2장 고르기 $\rightarrow nC_2$

Q. n이 10만 이상이라면? 1초 제한인 문제에서는 시간 초과!

기본 코드 2. 순열

P. N개의 정수로 이루어진 순열 중 사전순으로 M 번째 위치한 순열은?

모든 순열의 가짓수 -> $N!$

```
int n, m;
cin >> n >> m;
int cnt = 0; // count
vector<int> v(n);
for (int i = 0; i < n; i++) cin >> v[i];
do {
    if (cnt == m) {
        for (int i = 0; i < n; i++) {
            cout << v[i] << " ";
        }
        break;
    }
    cnt++;
} while (next_permutation(v.begin(), v.end()));
```

<algorithm> next_permutation 활용

기본 코드 2. 순열

P. N개의 정수로 이루어진 순열 중 사전순으로 M 번째 위치한 순열은?

모든 순열의 가짓수 $\rightarrow N!$

Q. n이 12 이상이라면? 1초 제한인 문제에서는 시간 초과!

백설 공주와 일곱 난쟁이

BOJ 3040번

9개의 정수가 주어졌을 때, 합이 100이 되는 7개의 정수를 찾는 문제

백설 공주와 일곱 난쟁이

BOJ 3040번

9개의 정수가 주어졌을 때, 합이 100이 되는 7개의 정수를 찾는 문제

7중 for문? ok!

```
vector<int>v(9);
for (int i = 0; i < 9; i++) cin >> v[i];
for (int n1 = 0; n1 < 9; n1++) {
    for (int n2 = n1 + 1; n2 < 9; n2++) {
        ...
        for (int n7 = n6 + 1; n7 < 9; n7++) {
            if (v[n1] + v[n2] + v[n3] + v[n4] + v[n5] + v[n6] + v[n7] == 100) {
                cout << v[n1] << "\n" << v[n2] << "\n" << v[n3] << "\n" <<
                v[n4] << "\n" << v[n5] << "\n" << v[n6] << "\n" << v[n7] << "\n";
                return 0;
            }
        }
    }
}
```

백설 공주와 일곱 난쟁이

BOJ 3040번

9개의 정수가 주어졌을 때, 합이 100이 되는 7개의 정수를 찾는 문제

전체 합에서 두 수를 빼 100을 만드는데 좋다!

```
for (int i = 0; i < 9; i++) {
    for (int j = i + 1; j < 9; j++) {
        if (sum - v[i] - v[j] == 100) {
            for (int k = 0; k < 9; k++) {
                if (k != i && k != j) cout << v[k] << "\n";
            }
            break;
        }
    }
}
```

모든 순열

BOJ 10974번

N자리 순열을 사전순으로 순서대로 출력하는 문제

모든 순열

BOJ 10974번

N자리 순열을 사전순으로 순서대로 출력하는 문제

<algorithm> next_permutation 활용!

<algorithm> prev_permutation 도 가능!

모든 순열

BOJ 10974번

추가

next_permutation을 이용해 조합 만들기

<https://www.acmicpc.net/problem/6603>

추첨상 사수 대작전! (Easy)

BOJ 20410번 (2020 Shake! 문제)

임의의 정수들이 주어졌을 때, 나머지 두 정수를 알아내 식을 완성하는 문제

추첨상 사수 대작전! (Easy)

BOJ 20410번

임의의 정수들이 주어졌을 때, 나머지 두 정수를 알아내 식을 완성하는 문제

주어 지는 정수 - $X1, X2, \text{Seed}, m$

제한 : $m \leq 100$ (소수), $0 \leq a, c < m$

주어 지는 식

-> (a, c) 조합의 개수는 최대 1만개!

$$X1 = (a \times \text{Seed} + c) \% m$$

$$X2 = (a \times X1 + c) \% m$$