

알튜비튜

브루트 포스

오늘은 '무식하게 풀기' 라고도 불리는 브루트 포스 알고리즘에 대해 배웁니다.
때로는 가장 쉬운 길이 정답일 때가 있죠.

“4자리 수의 암호로 된 **자물쇠**를 풀어보자.”

-> 0000 ~ 9999 까지 **모두 해보면** 된다.

브루트 포스

- 단순히 가능한 모든 경우를 찾는 기법
- 테스트 케이스의 크기에 따라 소요 시간이 엄청나게 커질 수 있음
- 하지만, 떠올리기 가장 쉬운 방법이므로 문제를 풀 때 가장 먼저 고려해야 하는 방법이기도 함
- 입력 범위와 시간복잡도를 잘 고려하여 선택하는 것이 중요

"100,000,000"

- 총 연산수가 약 1억 회 이하인 경우, 충분히 브루트 포스로 접근 가능
- 연산수 = 모든 경우의 수 * 각 경우의 연산 횟수
- 실제와는 거리가 있지만, 대략적인 브루트 포스 가능 기준으로 잡으면 유용함

-> 반복문, 비트마스크, 순열, 재귀함수 등으로 접근

/<> 2231번 : 분해합 - Bronze 2

문제

- n 의 분해합 = n 과 n 을 이루는 각 자리수의 합
- n 의 생성자 = 분해합이 n 인 어떤 자연수 m
- (예) 245의 분해합 = $245+2+4+5 = 256 \rightarrow 245$ 는 256의 생성자
- n 의 가장 작은 생성자 구하는 문제

제한 사항

- n 의 범위는 $1 \leq n \leq 1,000,000$

$\rightarrow n$ 이 최대 1,000,000 이므로 분해합을 구하는 연산 횟수의 최대는 대략
 $1,000,000 = 1,000,000 + 1 + 0 + 0 + 0 + 0 + 0 + 0 = 7$ 번

/<> 2231번 : 분해합 - Bronze 2

문제

- n 의 분해합 = n 과 n 을 이루는 각 자리수의 합
- n 의 생성자 = 분해합이 n 인 어떤 자연수 m
- (예) 245의 분해합 = $245+2+4+5 = 256 \rightarrow 245$ 는 256의 생성자
- n 의 가장 작은 생성자 구하는 문제

제한 사항

- n 의 범위는 $1 \leq n \leq 1,000,000$

$\rightarrow n$ 까지 돌리며 분해합 모두 계산해봐도, 최대 $1,000,000 * 7 < 1$ 억
이므로 브루트 포스 가능!

/<> 1436번 : 영화감독 슝 - Silver 5

문제

- 종말의 숫자: 어떤 수에 6이 적어도 3개 이상 연속으로 들어가는 수
- (예) 666, 1666, 2666, 3666, ..., 6661, ... 등
- N번째 종말의 숫자를 구하는 문제 (사전순)

제한 사항

- n의 범위는 $1 \leq n \leq 10,000$

제한 사항

- n 의 범위는 $1 \leq n \leq 10,000$

접근

- 666의 앞과 뒤에 숫자를 계속 붙이고, 오름차순 정렬
-> 충분히 가능! 다른 방법은?

제한 사항

- n 의 범위는 $1 \leq n \leq 10,000$

접근

- 666의 앞과 뒤에 숫자를 계속 붙이고, 오름차순 정렬
-> 충분히 가능! 다른 방법은?
- 666부터 차례로 666이 수에 있는지 판단 (시간초과 안나나?)
-> 10,000번째 종말의 숫자는 6,669,999 보다는 작다 (6,660,000 ~ 6,669,999: 1만 개)
-> $6,669,999 < 1\text{억}$ 이므로 충분히 브루트 포스 접근 가능!

제한 사항

- n 의 범위는 $1 \leq n \leq 10,000$

접근

- 666의 앞과 뒤에 숫자를 계속 붙이고, 오름차순 정렬
-> 충분히 가능! 다른 방법은?
- 666부터 차례로 666이 수에 있는지 판단 (시간초과 안나나?)
-> 10,000번째 종말의 숫자는 6,669,999 보다는 작다 (6,660,000 ~ 6,669,999: 1만 개)
-> $6,669,999 < 1\text{억}$ 이므로 충분히 브루트 포스 접근 가능!
- 더 효율적인 풀이가 존재할 수 있지만 시험장에서 단시간에 떠올리기 쉽지 않음
-> 브루트 포스 풀이는 비교적 떠올리기 쉬움
-> 일단 브루트 포스 생각해 보자! -> 시간 괜찮은지 확인 -> 바로 시도

/<> 1018번 : 체스판 다시 칠하기 - Silver 5

문제

- 검정(B) 또는 흰색(W) 으로 칠해져 있는 $n * m$ 보드
- 체스판은 검정, 흰색이 번갈아서 칠해져 있는 보드
- $n * m$ 보드에서 $8 * 8$ 크기의 체스판을 만드려 할 때, 다시 칠해야 하는 칸 수의 최솟값을 구하는 문제

제한 사항

- 입력 범위는 $8 \leq n, m \leq 50$

예제 입력

```
10 13
BBBBBBBBBWBWBW
BBBBBBBBBBWBWB
BBBBBBBBBWBWBW
BBBBBBBBBBWBWB
BBBBBBBBBWBWBW
BBBBBBBBBBWBWB
BBBBBBBBBWBWBW
BBBBBBBBBBWBWB
WWWWWWWWWWWWBWB
WWWWWWWWWWWWBWB
```

예제 출력

```
12
```

Hint

1. n, m 의 범위가 작네요!
2. 구하고자 하는 체스판의 크기가 $8 * 8$ 로 정해져 있네요..? 그럼 가능한 체스판의 경우는 무엇이 있죠? 얼마 없어보여요.

가능한 8 * 8 체스판

1. 흰색(w)으로 시작하는 경우

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |

2. 검은색(B)으로 시작하는 경우

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |

● 두 체스판은 정확히 반대!

제한 사항

- 입력 범위는 $8 \leq n, m \leq 50$

접근

- 가능한 체스판 경우는 2가지
- n, m 은 최대 50 이므로 $(0, 0)$ 인덱스부터 차례로 $8 * 8$ 체스판 만들 때 바꿔야 하는 칸 수 계산 -> 그 중 최솟값 구하기

-> $(50 - 8)^2 * 8^2 * 2 < 1\text{억}$ 이므로 접근 가능!

체스판 다시 칠하기 - 예제

↓

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B | B | B | B | B | B | B | B | W | B | W | B | W |
| B | B | B | B | B | B | B | B | B | W | B | W | B |
| B | B | B | B | B | B | B | B | W | B | W | B | W |
| B | B | B | B | B | B | B | B | B | W | B | W | B |
| B | B | B | B | B | B | B | B | W | B | W | B | W |
| B | B | B | B | B | B | B | B | B | W | B | W | B |
| B | B | B | B | B | B | B | B | W | B | W | B | W |
| B | B | B | B | B | B | B | B | B | W | B | W | B |
| W | W | W | W | W | W | W | W | W | W | B | W | B |
| W | W | W | W | W | W | W | W | W | W | B | W | B |



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |
| B | W | B | W | B | W | B | W |
| W | B | W | B | W | B | W | B |

체스판 다시 칠하기 - 예제

↓ (0, 0)

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B | W | B | W | B | W | B | W | W | B | W | B | W |
| W | B | W | B | W | B | W | B | B | W | B | W | B |
| B | W | B | W | B | W | B | W | W | B | W | B | W |
| W | B | W | B | W | B | W | B | B | W | B | W | B |
| B | W | B | W | B | W | B | W | W | B | W | B | W |
| W | B | W | B | W | B | W | B | B | W | B | W | B |
| B | W | B | W | B | W | B | W | W | B | W | B | W |
| W | B | W | B | W | B | W | B | B | W | B | W | B |
| W | W | W | W | W | W | W | W | W | W | B | W | B |
| W | W | W | W | W | W | W | W | W | W | B | W | B |

- 'B'로 시작하는 체스판을 만들기 위해 바꾸는 칸 수 = 'B'로 시작하는 체스판과 겹치치 않는 칸의 수
- 'W'로 시작하는 체스판을 만들기 위해 바꾸는 칸 수 = 'B'로 시작하는 체스판과 겹치는 칸의 수

체스판 다시 칠하기 - 예제

↓ (0, 5)

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B | B | B | B | B | B | W | B | W | B | W | B | W |
| B | B | B | B | B | W | B | W | B | W | B | W | B |
| B | B | B | B | B | B | W | B | W | B | W | B | W |
| B | B | B | B | B | W | B | W | B | W | B | W | B |
| B | B | B | B | B | B | W | B | W | B | W | B | W |
| B | B | B | B | B | W | B | W | B | W | B | W | B |
| B | B | B | B | B | B | W | B | W | B | W | B | W |
| B | B | B | B | B | W | B | W | B | W | B | W | B |
| W | W | W | W | W | W | W | W | W | W | B | W | B |
| W | W | W | W | W | W | W | W | W | W | B | W | B |

- 'B'로 시작하는 체스판을 만들기 위해 바꾸어야 하는 칸의 수 12
- 'W'로 시작하는 체스판을 만들기 위해 바꾸어야 하는 칸의 수:
64 (전체 체스판 칸 수) - 12 = 52
- 둘 중 최솟값 사용!

정리

- 모든 경우의 수를 다 해보는 브루트 포스
- 입력 범위가 커지는 만큼 시간초과 위험이 있음 (즉, 효율적인 방법은 아님)
- 그래도 비교적 떠올리기 쉽다
- 따라서, 연산의 횟수가 100,000,000 이하면 일단 시도하자!
- 설마가 사람 잡는다..

이것도 알아보세요!

- 브루트 포스를 재귀함수로 구현해 보아요 (종료 조건 중요)
- next_permutation과 비트마스크에 대해서도 알아보아요 (백트래킹 문제에서 많이 사용)

필수

- /<> 20055번 : 컨베이어 벨트 위의 로봇 - Silver 1
- /<> 2108번 : 통계학 - Silver 4

3문제 이상 선택

- /<> 2858번 : 기숙사 바닥 - Bronze 2
- /<> 14501번 : 퇴사 - Silver 4
- /<> 10448번 : 유레카 이론 - Bronze 2
- /<> 2503번 : 숫자 야구 - Silver 5
- /<> 1107번 : 리모컨 - Gold 5
- /<> 3077번 : 임진왜란 - Silver 3