

# 코드포스 8주차 문제 풀이

# 문제 난이도

	문제 번호	문제 이름	난이도
A	25400	제자리	B1
B	15786	Send me the money	B1
C	30892	상어 키우기	S1
D	31459	초콜릿과 7나이트 게임	S3
E	23633	소수 징글벨	G3
F	12904	A와 B	G5
G	1125	바닥 장식	P4
H	20136	멀티탭 스케줄링 2	P4

# A. 제자리

그리디 알고리즘

## A. 제자리

- ✓ 이 문제는 주어진 카드 배열에서 제자리 상태를 만들기 위해 최소 몇 장의 카드를 제거해야 하는지를 구하는 문제입니다.
- ✓ **필요한 값의 순서 찾기:**
  - ✓ 우리는 1부터 시작하여 연속적인 값이 나오는 순서를 찾고, 이를 제자리 카드로 간주합니다. 즉, 1, 2, 3, ... 순서로 나오는 최대 연속 부분 수열을 찾습니다.
- ✓ **제자리 카드의 개수 구하기:**
  - ✓ 위에서 찾은 최대 연속 부분 수열의 길이를 구합니다.
- ✓ **제자리 상태를 만들기 위한 최소 제거 카드 수 계산:**
  - ✓ 전체 카드 수에서 제자리 카드의 개수를 뺀 값이 최소 제거 카드 수가 됩니다.
- ✓ **시간 복잡도 :  $O(N)$**

## B. SEND ME THE MONEY

구현, 그리디 알고리즘, 문자열

## B. Send me the money

- ✓ 이 문제는 석규가 기억하는 비밀번호의 일부 문자열이 주어졌을 때, 여러 포스트잇 중에서 해당 문자열이 순서대로 포함되는지를 판별하는 문제입니다.
- ✓ 포스트잇 문자열을 왼쪽에서 오른쪽으로 순차적으로 탐색하며, 석규가 기억하는 비밀번호 문자열  $S$ 의 각 문자가 순서대로 나타나는지 확인합니다.
- ✓  $S$ 의 첫 번째 문자부터 시작하여 포스트잇 문자열에서 일치하는 문자를 찾으면 다음 문자로 넘어갑니다.
- ✓ 만약  $S$ 의 모든 문자를 포스트잇 문자열에서 순서대로 찾았다면, 그 포스트잇 문자열은 비밀번호일 가능성이 있습니다.
- ✓ 시간 복잡도 :  $O(ML)$

## C. 상어 키우기

자료 구조, 그리디 알고리즘, 스택, 정렬, 분리 집합

## C. 상어 키우기

- ✓ 이 문제는 아기 상어가 최대 K마리의 작은 상어를 먹어 몸집을 최대화하는 문제입니다.
- ✓ **정렬:**
  - ✓ 먼저 상어의 크기 배열을 정렬합니다. 이렇게 하면 작은 크기의 상어부터 순차적으로 접근할 수 있습니다.
- ✓ **이진 탐색 및 Union-Find:**
  - ✓ 매번 아기 상어 T가 먹을 수 있는 가장 큰 상어를 이진 탐색을 통해 찾습니다. 이를 위해 이진 탐색을 사용합니다.
  - ✓ 이진 탐색을 사용하면 T 이상인 첫 번째 상어의 위치를 가져오므로 그 앞의 상어가 가장 큰 먹을 수 있는 상어가 됩니다.
  - ✓ Union-Find 자료구조를 사용하여 이미 먹힌 상어를 관리합니다. Find함수는 상어의 부모를 찾고, 상어가 먹히면 부모를 업데이트합니다.



## C. 상어 키우기

- ✓ **반복:**
  - ✓ 최대 K번 상어를 먹는 과정을 반복합니다.
  - ✓ 만약 먹을 수 있는 상어가 더 이상 없으면 반복을 종료합니다.
  - ✓ 매번 먹을 수 있는 상어를 먹고, 아기 상어 T의 크기를 증가시킵니다.
- ✓ **시간 복잡도:**
  - ✓ 정렬 :  $O(N\log N)$
  - ✓ 이진 탐색 :  $O(\log N)$
  - ✓ Union-Find : 각 상어를 먹을 때  $O(\log N)$
  - ✓ 전체 시간 복잡도 :  $O(N\log N) + O(K\log N)$

## D. 초콜릿과 $\neg$ 나이트 게임 (SWEET)

그리디 알고리즘

## D. 초콜릿과 ㄱ나이트 게임 (Sweet)

### ✓ 탐색 및 배치:

- ✓ 초콜릿의 각 칸을 순회하면서, 아직 방문하지 않은 칸에 대해 ㄱ나이트를 배치합니다.
- ✓ ㄱ나이트를 배치할 때마다 그 위치를 방문했다고 표시하고, 카운트를 증가시킵니다.
- ✓ ㄱ나이트가 이동할 수 있는 위치를 계산하고 그 위치를 방문할 수 없도록 표시합니다.

### ✓ 결과 출력:

- ✓ 각 테스트 케이스에 대해 최대 몇 개의 ㄱ나이트를 배치할 수 있는지 출력합니다.

### ✓ 시간 복잡도:

- ✓ 각 테스트 케이스마다 초콜릿의 모든 칸을 순회하므로, 시간 복잡도는  $O(TXY)$ 입니다.

## E. 소수 징글벨

수학, 다이나믹 프로그래밍, 그리디 알고리즘, 정수론, 게임 이론, 소수 판정

## E. 소수 징글벨

### ✓ 소수 판별 배열 생성:

- ✓ 최대 2000까지의 소수 여부를 미리 계산하여 배열에 저장합니다. (prime이라고 합니다.) 이 배열은 에라토스테네스의 체를 사용하여 생성됩니다.

### ✓ 테스트 케이스 처리:

- ✓ 현재 인덱스(idx)부터 갈 수 있는 최대 인덱스(idx + A) 까지의 범위 내에서 징글벨을 칩니다.
- ✓ 범위 내에서 소수 번째 징글벨 횃수를 찾아 해당 플레이어의 점수를 증가시킵니다.
- ✓ 가장 마지막에 친 소수 번째 징글벨 횃수를 현재 인덱스(idx)로 업데이트 하고 턴을 바꿉니다.
- ✓ 게임이 종료된 후, 점수를 비교하여 승자를 출력합니다.

### ✓ 시간 복잡도: $O(TB)$

## F. A와 B

구현, 그리디 알고리즘, 문자열

## F. A와 B

### ✓ 기본 아이디어:

- ✓ 우리는 주어진 두 문자열 S와 T에 대해 T를 S로 변환하는 과정을 거꾸로 생각해봅시다. 즉 T에서 시작하여 S로 변환 가능한지를 판단합니다.
- ✓ 마지막 문자가 A인 경우 A를 제거합니다.
- ✓ 마지막 문자가 B인 경우 B를 제거하고 문자열을 뒤집습니다.

### ✓ 과정:

- ✓ T를 계속 변환하여 S와 길이가 같아질 때까지 반복합니다.
- ✓ 각 단계에서 T의 마지막 문자를 확인하고, 위의 규칙에 따라 변환을 수행합니다.
- ✓ 변환이 끝난 후 S와 T가 같다면 1, 아니면 0을 출력합니다

### ✓ 시간 복잡도: $O(T)$

## G. 바닥 장식

수학, 그리디 알고리즘, 많은 조건 분기



## G. 바닥 장식

### ✓ 필요한 타일 조각들 계산:

- ✓ X범위가 2 ~ 4 같은 5X5 그리드 안에 들어와 있는 경우, 3 ~ 6 같은 패턴이 바뀌는 경계에 걸쳐 있는 경우, 9 ~ 101 같이 5X5 그리드를 통째로 포함하는 경우가 있습니다.
- ✓ 각각의 경우를 따로 계산해줘야 하는데, 이 때의 Y범위도 위와 같이 분리하여 구해줘야 합니다.
- ✓ 좌표가 5의 배수인 지점에서 패턴이 바뀌기 때문에, 이를 고려해야 합니다.
- ✓ 예컨대, X 범위가 4 ~ 13이라면 4~5, 5~10, 10~13이런 식으로 따로 계산해야 하며, 이 때의 Y 범위도 마찬가지로 따로 계산하여 구해줘야 합니다. (πππ)

### ✓ 사야 할 최종 1X5 타일 수 계산:

- ✓ 1X5 타일을 정답에 더해줍니다.
- ✓ 1X4 타일과 1X1 타일을 1:1 비율로 계산하여 정답에 더해줍니다.
- ✓ 1X3 타일과 1X2 타일을 1:1 비율로 계산하여 정답에 더해줍니다.
- ✓ 1X3 타일과 1X1 타일을 1:2 비율로 계산하여 정답에 더해줍니다.
- ✓ 1X2 타일과 1X1 타일을 2:1 비율로 계산하여 정답에 더해줍니다.
- ✓ 1X2 타일과 1X1 타일을 1:3 비율로 계산하여 정답에 더해줍니다.
- ✓ 1X1 타일을 5개 묶음으로 정답에 더해줍니다.

## H. 멀티탭 스케줄링 2

자료 구조, 그리디 알고리즘, 우선순위 큐

## H. 멀티탭 스케줄링 2

- ✓ 이 문제는 멀티탭 스케줄링 문제로, 주어진 전기 용품의 사용 순서에 따라 멀티탭에서 플러그를 빼는 최소 횟수를 구하는 문제입니다.
- ✓ **사용 순서 기록:**
  - ✓ 각 전기 용품의 사용 순서를 기록하면서, 해당 용품이 마지막으로 사용된 인덱스를 배열에 저장합니다. 이렇게 하면 각 용품이 다음에 사용될 때 어느 시점에 다시 꽂아야 하는지를 알 수 있습니다.
- ✓ **우선순위 큐 활용:**
  - ✓ 현재 멀티탭에 꽂혀 있는 용품들 중에서 가장 나중에 사용될 용품을 먼저 빼야 최소 횟수로 빼는 경우를 찾을 수 있습니다. 이를 위해 우선순위 큐(priority queue)를 사용합니다.

## H. 멀티탭 스케줄링 2

- ✓ 우선순위 큐에는 현재 멀티탭에 꽂혀 있는 용품들 중, 가장 나중에 사용될 용품의 정보를 저장합니다. 이를 위해 pair를 사용하여 (사용되는 시점, 용품 번호)를 저장합니다.
- ✓ 멀티탭에 더 이상 꽂을 자리가 없을 때마다 우선순위 큐에서 가장 나중에 사용될 용품을 빼고, 이 때 빼는 횟수를 카운트합니다.
- ✓ 시간 복잡도:  $O(K \log N)$