

코드포스 2주차 문제 풀이

문제 난이도

	문제 번호	문제 이름	난이도
A	17614	369	B3
B	28239	배고파(Easy)	B1
C	28447	마라탕 재료 고르기	S2
D	26595	전투의 신	S5
E	20008	몬스터를 처치하자!	G5
F	3363	동전	G5
G	30895	쇼진의 창	P5
H	26600	map, filter	P4

A. 369

구현, 브루트포스 알고리즘

A. 369

- ✓ 기존의 369룰을 똑같이 가져왔습니다. 숫자 안에 3이나 6이나 9가 있으면 있는 만큼 카운트 해서 출력합니다.
- ✓ 시간복잡도 : $O(N)$

B. 배고파(EASY)

수학, 브루트포스 알고리즘, 비트마스킹

B. 배고파(Easy)

- ✓ m을 이진수로 나타냈을 때의 비트가 2개이거나 1개입니다.
- ✓ 2개인 경우 작은것 부터 차례로 출력하고, 1개인 경우 두 번 출력하면 정답입니다.
- ✓ 시간복잡도 : $O((\lg M)^2)$

C. 마라탕 재료 고르기

브루트포스 알고리즘, 백트래킹

C. 마라탕 재료 고르기

- ✓ N개 중 K개를 골라 조합하는 문제입니다. N이 최대 10이고 K의 값도 1~N 이므로 가능한 조합을 전부 구해서 조합의 맛의 총합의 최대값을 구하는 문제입니다.
- ✓ 시간복잡도 : $O(N^K)$

D. 전투의 신

브루트포스 알고리즘

D. 전투의 신

- ✓ 탱커를 한 명도 고용하지 않는 경우부터 예산 N 을 모두 사용해서 탱커만 고용하는 경우까지 고려합니다.
- ✓ 탱커를 x 명 고용하는 경우의 비용은 $x * P_a$ 입니다.
- ✓ 남은 예산으로 딜러를 최대한 많이 고용합니다.
- ✓ 딜러를 고용하는 비용은 $y * P_b$ 입니다.
- ✓ 이 때의 전투력은 $Ax + By$ 입니다. 전투력이 최대가 되는 경우를 선택합니다.
- ✓ 시간복잡도 : $O(\frac{N}{P_A})$

E. 몬스터를 처치하자!

브루트포스 알고리즘, 백트래킹

E. 몬스터를 처치하자!

- ✓ 이 문제는 여러 가지 스킬을 사용하여 주어진 몬스터의 체력을 가장 빠르게 0 이하로 만드는 최소 시간을 찾는 문제입니다.
- ✓ 각 스킬은 대기 시간이 다르며, 동일한 스킬을 연속으로 사용하기 위해서는 일정 시간이 지나야 합니다.
- ✓ 이 문제를 해결하기 위해 재귀함수 사용해볼 수 있습니다. 재귀함수를 사용하여 모든 가능한 스킬 사용 시퀀스를 탐색하고, 몬스터의 체력이 0 이하가 되는 최소 시간을 찾습니다.
- ✓ 시간복잡도 : $O(N^N)$

F. 동전

구현, 브루트포스 알고리즘

F. 동전

- ✓ 이 문제는 12개의 동전 중 하나가 모조품임을 찾아내는 문제입니다. 모조품 동전은 다른 동전보다 무겁거나 가볍습니다.
- ✓ 일관성 검사
 - ✓ 동전이 무겁거나 가벼운 가정하에, 각 저울 측정 결과를 확인하여 해당 동전이 모조품일 가능성을 평가합니다.
 - ✓ 예를 들어, 동전 n 이 무겁다고 가정했을 때, 각 측정에서 이 가정이 저울 결과와 맞는지 검사합니다.
 - ✓ 마찬가지로, 동전 n 이 가볍다고 가정한 경우도 검사합니다.

F. 동전

✓ 결과 도출

- ✓ 일관성이 있는 모조품 동전의 가정을 모아서 결과를 출력합니다.
- ✓ 만약 가능한 가정이 하나도 없으면 "impossible"을 출력합니다.
- ✓ 가능한 가정이 여러 개 있으면 "indefinite"을 출력합니다.
- ✓ 가능한 경우가 하나만 있으면 그 경우를 출력합니다.
- ✓ 시간복잡도 : $O(12 \times 2 \times 3 = 72) = O(1)$

G. 쇼진의 창

수학, 그리디 알고리즘, 브루트포스 알고리즘

G. 쇼진의 창

- ✓ 이 문제는 챔피언이 스킬을 사용하기 위해 필요한 기본 공격 횟수를 계산하여, 효율이 가장 낮은 최대 마나 M 을 찾는 문제입니다.
- ✓ 효율이 낮아질 때는 M 이 $(X + Y)$ 의 배수 + 1일 때 입니다. 그렇기 때문에, M 을 탐색할 때 $X+Y$ 만큼 건너뛰면서 탐색하면 효율적으로 답을 알아낼 수 있습니다.
- ✓ $K + (X + Y) * X$ 까지 탐색한다면 최적의 해를 구할 수 있습니다.
- ✓ 시간복잡도 : $O(\frac{K}{X + Y})$

H. MAP, FILTER

수학, 구현, 브루트포스 알고리즘, 정렬, 이분 탐색, 많은 조건 분기

H. map, filter

- ✓ map 연산의 +와 *는 일일이 배열에 연산하지 않고, 초기값을 $a = 1, b = 0$ 으로 둔 뒤에 +연산은 b 에 더하고, *연산은 a 와 b 에 모두 곱해서 저장합니다.
- ✓ filter 연산시에 피연산자에 b 를 빼고 a 만큼 나누어 진행할 수 있습니다.
- ✓ map 연산의 %는 피연산자의 최대 범위가 100이므로 일일이 배열에 연산해서 누적합으로 구현해볼 수 있습니다.
- ✓ filter 연산할 때 배열을 미리 정렬시키고 이분탐색으로 답을 구해서 출력합니다. (약간의 case-work 필요)
- ✓ 시간복잡도 : $O(Q \lg N)$