

생각하는 프로그래밍 해설

문제	난이도	출제자
A 고무오리 디버깅	Easy	최민우 (dandalf@Tistory, cmw0107@BOJ)
B 사과나무	Easy	복신영(Sinyoung3016@GitHub, sin0415@BOJ)
C 거스름돈이 싫어요	Easy	정지수 (Jeesu-Jung@GitHub, bluedao@BOJ)
D 베스킨라빈스 31	Medium	강시온 (Yaminyam@GitHub, siontama@BOJ)
E 보스몬스터 전리품	Medium	김종운 (potionk@GitHub, potion@BOJ)
F 랭킹전 대기열	Medium	김종운 (potionk@GitHub, potion@BOJ)
G 떡돌리기	Medium	강시온 (Yaminyam@GitHub, siontama@BOJ)
H 몬스터를 처치하자!	Medium	정지수 (Jeesu-Jung@GitHub, bluedao@BOJ)
I 형곤이의 소개팅	Hard	유형곤 (hygoni@GitHub, hygoni@BOJ)
J 악덕 영주 혜유	Hard	유형곤 (hygoni@GitHub, hygoni@BOJ)

Special thanks to

내부 검수진
protoseo
shinwoon
b5460881
willook
외부 검수진
isku
amok
sait2000

A. 고무오리 디버깅

Stack

Easy

A. 고무오리 디버깅

스택을 하나 생성해 문제가 들어오면 스택에 추가

고무오리가 들어오면 스택에서 제거

하지만 스택이 비어있을시, 스택에 2개 추가

B. 사과나무

DP

Easy

B. 사과나무

(0, 0)부터 (i, j)칸까지의 합을 $arr[i, j]$ 라 할 때,

$arr[i, j] = arr[i - 1][j] + arr[i][j - 1] - arr[i - 1][j - 1] + map[i][j]$ 이라고 할 수 있다

그리고 n의 크기의 정사각형 모양으로 수확해야 하므로 (i, j)칸을 왼쪽 위 꼭지점으로 하는 정사각형의 이익은

$value = arr[i + n][j + n] + arr[i][j] - arr[i + n][j] - arr[i][j + n]$

으로 구할 수 있다

C. 거스름돈이 싫어요

Math

Easy

C. 거스름돈이 싫어요

먼저 모든 분수를 기약분수로 만들기 위해 분자, 분모의 최대공약수를 구하여 각각 나눠준다

새로운 코인 단위에 n 을 곱하여 모든 아이템가격의 분모를 만들 수 있어야 하므로
모든 분모들의 최소공배수를 A

가장 큰 값을 찾아야 하므로 새로운 코인 단위의 분자에 n 을 곱하여 모든 아이템가격의 분자를 만들 수 있는
모든 분자들의 최대공약수를 B

새로운 코인 단위 B / A

D. 베스킨라빈스 31

Sprague Grundy

Medium

D. 베스킨라빈스 31

보통 베스킨라빈스게임의 필승법은 4-상대방이 부른 숫자의 갯수 만큼을 부르는 것이다

이 게임의 원리는 게임이론 스프라그-그런디 정리를 이용하면

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	31
L	W	W	W	L	W	W	W	L	W	W	W	L	W	W	...	W

선공이 지는 상황이 만들어지는 숫자의 남은 갯수는 $n \% 4 == 1$ 일 때 인 것을 알아 낼 수 있다

D. 베스킨라빈스 31

또 3개가 아닌 4, 5개의 수를 부를 수 있을 때의 경우도 알아보면

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	31
L	W	W	W	W	L	W	W	W	W	L	W	W	W	W	...	L

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	31
L	W	W	W	W	W	L	W	W	W	W	W	L	W	W	...	L

31개의 숫자가 남았을 때 지는 상황으로 시작을 해야 선공이 지는 상태이므로 시온이가 게임을 이길 수 있게된다

여기에서 규칙을 찾게 되면 31 % (n + 1) == 1인 n을 찾으면 시온이가 게임을 이길 수 있을 것이다

E. 보스몬스터 전리품

BFS

Medium

E. 보스몬스터 전리품

공격과 이동 모두 1초가 소모 되는 행동이기 때문에 queue에 플레이어들을 넣어주어 BFS로 이동이나 공격을 한 번씩 처리 하여 문제를 해결한다

BFS를 이용하여 모든 플레이어들이 보스에게 도달 할 수 있는 최단 경로로 이동한다

처음으로 보스에게 도달 되었다면 정답을 1 증가 시켜주고 보스에게 도달 된 플레이어는 보스를 공격한다

단, 모든 캐릭터의 행동은 동시에 일어나지만 BFS의 처리는 한 플레이어씩 이루어 지기 때문에 보스와의 전투 종료는 queue에 있는 모든 플레이어들의 행동을 전부 마친후에 종료된다

F. 랭킹전 대기열

Implementation

Medium

F. 랭킹전 대기열

문제의 주어진 조건대로 구현한다면 풀 수 있는 문제이다

리스트나 벡터같은 순서가 있는 자료구조를 이용하여 방과 방들을 구현하고 처음 입장한 플레이어의 레벨을 기준으로 +- 10레벨의 플레이어들을 입장시킨다

순서가 있는 자료구조를 이용하면 처음 입장한 플레이어의 레벨을 쉽게 알 수 있고 방을 생성된 순서대로 출력 할 수 있다

G. 떡 돌리기

Dijkstra

Medium

G. 떡 돌리기

먼저, 다익스트라 알고리즘을 이용하여 모든 이웃집들의 최단거리를 구한다

성현이는 한 번에 한 개의 떡만 들고 갈 수 있고 잠은 집에서 자야하므로 무조건 이동은 왕복을 해야하므로 떡을 전달하는 이웃집과의 코스트는 최단거리 $\times 2$ 라고 할 수 있다

떡 전달 코스트들을 오름차순으로 정렬한 뒤 성현이가 하루에 갈 수 있는 거리인 X 이하의 수의 합으로 이웃집들을 나누어 주면 몇일이 걸리는지 알 수 있다

H. 몬스터를 처치하자!

Backtracking

Medium

H. 몬스터를 처치하자!

무조건 공격력이 썬 스킬을 시전하는 것으로 문제는 풀 수 없다는 점을 유의해야한다
또한, 모든 조합을 만들어 볼 시 시간초과

가용가능한 스킬이 없을시 시간을 +1

가용가능한 스킬이 1개면 무조건 실행

가용가능한 스킬이 여러개면 DFS로 분기해서 탐색하여 최소값을 구한다

I. 형곤이의 소개팅

Gale-Shapley

Hard

I. 형곤이의 소개팅

이 문제는 Stable marriage problem으로 널리 알려진 문제이다

남자들의 선호도가 가장 높은 여자와 임시로 짝이 된다

이미 매칭이 된 여자를 남자가 선택 할 경우 여자가 가장 선호하는 남자를 골라 다시 임시로 짝이 된다

아직 짝이 없는 남자는 그 다음 선호도의 여자를 선택하고 여자의 임시짝이 있다면 여자의 선호도가 더 높은 남자와 다시 임시로 짝이 된다

모든 짝이 연결될 때 까지 반복하면 최적의 짝을 찾을 수 있다

J. 악덕 영주 혜유

MST

Hard

J. 악덕 영주 혜유

모든 마을 을 갈 수 있도록 교역로를 연결 할 때, 최소한의 비용으로 연결을 하게 되면 그래프 내의 모든 정점을 연결 하면서 최소의 비용을 가지는 Minimum Spanning Tree의 형태가 된다.

MST는 Kruskal 알고리즘을 이용하여 만들 수 있다

그래프 내의 모든 간선을 오름차순으로 정렬하고 가중치가 작은 간선부터 선택하고 Union Find 기법을 사용하여 이미 같은 트리내에 존재하는 간선의 연결을 배제해 회로가 생기지 않도록 하면 MST를 만들 수 있다

MST의 모든 가중치의 합을 구하면 모든 마을을 연결하는 최소 비용을 구할 수 있다

마을과 마을 을 이동하는 가장 큰 비용의 경로의 비용은 MST를 DFS를 이용해 탐색해서 트리의 지름을 구하여 알아 낼 수 있다