

CPC 2022 div2 풀이

Official Solutions

중앙대학교 알고리즘 학회 ChAOS

Div 2	문제명	난이도
A	푸앙이와 종윤이	Easy
B	바둑돌 게임	Easy
C	근무 지옥에 빠진 푸앙이 (Small)	Medium
D	HG 음성기호	Medium
E	86 -에이티식스- 2	Medium
F	마트료시카 박스 II	Hard
G	푸앙이와 러닝머신	Hard
H	Alive or Dead?	Hard

A. 푸앙이와 종윤이

arithmetic, mathematics

출제진 의도-**Easy**

 출제자: BothEarRim

A. 푸양이와 종윤이

- ✓ 문제에서 a, b, c, d, q, r 에 대한 설명을 친절하게 하고 있어 찬찬히 읽고 그대로 값들을 구하면 됩니다.

B. 바둑돌 게임

implementation, mathematics, simulation

출제진 의도 - **Easy**

✅ 출제자: saywoo

B. 바둑돌 게임

- ✓ 바둑돌의 최대 개수가 100,000개이므로 $\frac{x \times (x+1)}{2} \leq 100,000$ 을 만족하는 가장 큰 자연수 x 는 약 450입니다.
- ✓ 그러므로 이러한 사실을 알 수 있습니다.
 - ✓ 450번째 차례 안에 항상 게임이 끝나게 됩니다.
 - ✓ 450개 이하의 바둑돌을 추가했을 때, 항상 푸앙이가 게임에서 이기는 방법이 존재합니다.

B. 바둑돌 게임

- ✓ 이 게임을 수행하는 시뮬레이터를 작성합니다.
- ✓ 바둑돌을 0개 더했을 때, 바둑돌을 1개 더했을 때, 바둑돌을 2개 더했을 때, ..., 바둑돌을 450개 더했을 때를 모두 시뮬레이터를 이용해 게임을 수행합니다.
- ✓ 푸양이가 게임을 이기는 경우 중, 더한 바둑돌의 개수가 가장 적을 때가 답이 됩니다.

C. 근무에 빠진 푸앙이 (Small)

implementation, string

출제진 의도 - **Medium**

✅ 출제자: smmaker118

C. 근무에 빠진 푸앙이 (Small)

- ✓ N의 최대 값이 50이므로 근무표에 존재하는 이름의 수는 $50 \times 7 \times 4$ 로 최대 1400개입니다.
- ✓ 근무자의 수는 제시된 것처럼 최대 100명입니다.(이때 근무자 수를 M이라고 해봅시다.)
- ✓ 근무자의 이름의 길이는 제시된 것처럼 최대 20입니다.

C. 근무에 빠진 푸앙이 (Small)

- ✓ 근무자의 이름과 근무 시간을 저장하는 배열을 만들어 근무표를 탐색하면서 근무 시간을 계산할 수 있습니다.
- ✓ 최대 근무 시간과 최소 근무시간을 비교하여 근무표의 공정성을 확인 할 수 있습니다.
- ✓ 풀이대로 구현하면 총 시간복잡도 $O(NM)$ 에 해결할 수 있습니다.

D. HG 음성기호

bruteforcing, case_work, string
출제진 의도 - **Medium**

✅ 출제자: xingxing2001

D. HG 음성기호

- ✓ 입력된 문자열이 HG 음성기호 26개 중 하나로 이루어져 있는지 전부 체크하면 되는 문제입니다.
- ✓ 입력된 문자열을 검사할 때 해당 단어가 검사할 문자열의 시작인지 아닌지를 체크한 뒤, 문자열의 시작이면 A~Z에 해당하는 HG 음성기호 단어와 비교를 하면 됩니다.
- ✓ 위에서 말한 방법을 사용하면 $O(N)$ 시간복잡도로 해결 가능합니다

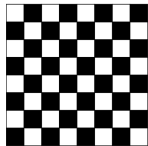
E. 86 —에이티식스— 2

ad_hoc

출제진 의도 - **Medium**

 출제자: BothEarRim

E. 86-에이티식스-2




- ✓ 체스판처럼 검은색, 흰색 칸으로 영역을 나눠 생각해보면
신에이 노우젠은 같은 색의 영역만 갈 수 있습니다.
- ✓ 그런데, 레기온을 해치우는 경우는 다른 색의 영역에 있는 레기온들 뿐입니다.
- ✓ 따라서, 모든 레기온이 신에이 노우젠과 다른 색의 영역에만 있는지 검사하면 됩니다.
- ✓ 특정 좌표가 어느 색의 영역인지 판단하는 방법은 간단하게 좌표의 행+열 값이 홀/짝인지로 구분하면 됩니다.

F. 마트료시카 박스 II

greedy

출제진 의도 - **Hard**

 출제자: wapas

F. 마트로시카 박스 II

- ✓ 박스를 낭비 없이 새로운 박스를 추가하려면, 서브 박스가 M개 초과인 박스에만 새로운 박스를 추가해야 합니다.
- ✓ i 번 박스의 서브 박스 개수를 s_i 라고 정의합시다.
- ✓ 그리고 모든 박스에 대해 한 번씩 아래 과정을 진행합니다.
 - ✓ s_i 가 M 이하라면 다음 박스를 봅니다.
 - ✓ s_i 가 M 초과라면 새로운 박스를 생성하고, $s_i - M + 1$ 개의 박스를 새로운 박스의 서브 박스로 옮깁니다.
- ✓ 박스를 추가하는 과정에서 K개를 초과하여 박스를 추가했다면 0을 출력합니다.

G. 푸양이와 러닝머신

greedy

출제진 의도 - **Hard**

 출제자: halin

G. 푸앙이와 러닝머신

- ✓ 주어진 거리 X 를 최단 시간에 완주하고자 할 때의 해 x, y, z 를 그리디하게 구할 수 있습니다.
- ✓ $X = 8x + 4y + z$ 라 할 때, 목표한 시간 T 가 $x + y + z$ 보다 작으면 시간 T 안에 완주가 불가능합니다.
- ✓ 한편 누르는 버튼의 종류를 최소화하고 싶기 때문에,

$$\min(x + 4y + z, 2x + y + z)$$

이상의 시간이 주어지면 세 종류의 버튼을 누르는 대신 두 종류의 버튼을 누를 수 있습니다.

G. 푸앙이와 러닝머신

- ✓ 마찬가지로 두 종류의 버튼을 누르는 대신, 계수가 더 큰 해를 계수가 더 작은 해로 옮겨 한 종류의 버튼을 누를 수 있습니다.
- ✓ 주어진 시간 T 에 누르는 버튼의 종류가 최소가 되도록 하는 해를 찾아, T 에 완주할 수 있도록 출력하면 됩니다.

H. Alive or Dead?

implementation, simulation

출제진 의도 - **Hard**

✅ 출제자: xingxing2001, wapas, hyeseungmoon

H. Alive or Dead?

- ✓ 플레이어의 위치는 py (행), px (열)로 선언 뒤, 하나의 변수로 생각하고 이동 처리를 해도 됩니다.
- ✓ 벽의 위치는 최대 26×26 배열(좌표처리를 편하게 하기 위해)을 선언한 뒤 원소들을 전부 0으로 초기화시킨 뒤
- ✓ 입력된 벽의 위치 wy (행), wx (열)에 해당하는 원소를 1로 변경하여 벽의 유무를 나타낼 수 있습니다.

H. Alive or Dead?

- ✓ 그렇다면 좀비들을 어떻게 처리하면 될까요?
- ✓ 좀비의 정보를 일반적인 변수들로 하나하나 선언하면 될까요??
- ✓ 그렇지 않습니다. 문제에서의 좀비는 최대 624마리나 주어질 수 있기 때문입니다.

H. Alive or Dead?

- ✓ 좀비의 이동처리를 하려면 구조체 혹은 객체를 활용하여 좀비의 정보를 담은 객체 혹은 변수를 하나씩 선언합니다.
- ✓ 그런 뒤 문제에서 주어진 좀비의 이동 방식에 맞게끔 구현을 하면 됩니다.
- ✓ 하급 좀비는 단순 반복문을 사용해 L번 돌려도 되지만, 상급 좀비는 H번 돌리면 될까요? 안됩니다.
- ✓ 그래서 상급 좀비의 이동속도와 현재 갈 수 있는 칸을 비교하여 현재 갈 수 있는 칸이 이동속도보다 적다면 현재 갈 수 있는 칸만큼만 반복문을 돌리면 시간초과를 피할 수 있습니다.

H. Alive or Dead?

- ✓ 이런 식으로 플레이어 이동 -> 좀비 이동을 반복한 뒤
- ✓ 모든 이동이 끝나면 좀비들과 플레이어의 위치가 일치하는지 아닌지를 계속 검사하면 됩니다.