

[Algorithm] 15강 : 구현 유형 문제 풀이 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-28 오후 11:25

업데이트: 2020-10-28 오후 11:30

URL: <https://continuous-development.tistory.com/163>

Algorithm

[Algorithm] 15강 : 구현 유형 문제 풀이

2020. 10. 28. 08:04 수정 삭제 공개

문제:시각

〈문제〉 시각: 문제 설명

- 정수 N이 입력되면 00시 00분 00초부터 N시 59분 59초까지의 모든 시각 중에서 3이 하나라도 포함되는 모든 경우의 수를 구하는 프로그램을 작성하세요. 예를 들어 1을 입력했을 때 다음은 3이 하나라도 포함되어 있으므로 세어야 하는 시각입니다.
 - 00시 00분 03초
 - 00시 13분 30초
- 반면에 다음은 3이 하나도 포함되어 있지 않으므로 세면 안 되는 시각입니다.
 - 00시 02분 55초
 - 01시 27분 45초

〈문제〉 시각: 문제 조건

난이도 ●○○ | 풀이 시간 15분 | 시간제한 2초 | 메모리 제한 128MB

입력 조건 • 첫째 줄에 정수 N이 입력됩니다. ($0 \leq N \leq 23$)

출력 조건 • 00시 00분 00초부터 N시 59분 59초까지의 모든 시각 중에서 3이 하나라도 포함되는 모든 경우의 수를 출력합니다.

입력 예시

5

출력 예시

11475

문제 해결 아이디어

이 문제는 모든 시간의 경우를 하나씩 모두 세서 풀 수 있는 문제이다.

하루는 86,400초이므로 파이썬이 1초에 20만 계산을 할 수 있는 걸 생각해 완전 탐색으로 진행하면 된다.

#코드

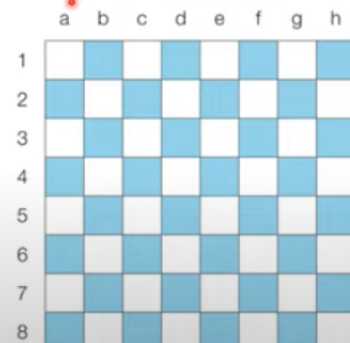
```
# H 입력받기
h = int(input())

count = 0
for i in range(h+1):
    for j in range(60):
        for k in range(60):
            # 매 시간안에 3 이 포함되어 있다면 카운트 증가
            if '3' in str(i) + str(j) + str(k):
                count += 1
print(count)
```

문제 : 왕실의 나이트

〈문제〉 왕실의 나이트: 문제 설명

- 행복 왕국의 왕실 정원은 체스판과 같은 8×8 좌표 평면입니다. 왕실 정원의 특정한 한 칸에 나이트가 서 있습니다. 나이트는 매우 충성스러운 신하로서 매일 무술을 연마합니다.
- 나이트는 말을 타고 있기 때문에 이동을 할 때는 L자 형태로만 이동할 수 있으며 정원 밖으로는 나갈 수 없습니다.
- 나이트는 특정 위치에서 다음과 같은 2가지 경우로 이동할 수 있습니다.
 - 수평으로 두 칸 이동한 뒤에 수직으로 한 칸 이동하기
 - 수직으로 두 칸 이동한 뒤에 수평으로 한 칸 이동하기



〈문제〉 왕실의 나이트: 문제 조건

난이도 ●○○ | 풀이 시간 20분 | 시간 제한 1초 | 메모리 제한 128MB

입력 조건

• 첫째 줄에 8×8 좌표 평면상에서 현재 나이트가 위치한 곳의 좌표를 나타내는 두 문자로 구성된 문자열이 입력된다. 입력 문자는 a1처럼 열과 행으로 이뤄진다.

출력 조건

• 첫째 줄에 나이트가 이동할 수 있는 경우의 수를 출력하시오.

입력 예시

a1

출력 예시

2

문제 해결 아이디어

별 다른건 없고 나이트가 움직일 수 있는 경로를 모두 확인하면 된다.

총 8가지 경로가 있고 그 위치로 이동이 가능한지 확인하면 된다.

이때 리스트를 이용하여 8가지 방향에 대해 방향벡터를 정의해서 문제를 풀어준다.

코드

```
# 현재 나이트의 위치 입력받기
input_data = input()
row = int(input_data[1])
column = int(ord(input_data[0])) - int(ord('a')) + 1 # 아스키 코드값으로 1~9 까지 만듬

# 나이트가 이동할 수 있는 8 가지 방향 정의
steps = [(-2,-1),(-1,-2),(1,-2),(-2,-1),(2,1),(1,2),(-1,2),(-2,1)]

# 8 가지 방향에 대하여 각 위치로 이동이 가능한지 확인
result = 0
for step in steps:
    # 이동하고자 하는 위치 확인
    next_row = row + step[0]
    next_column = column + step[1]
    # 해당 위치로 이동이 가능하다면 카운트 증가
    if next_row >= 1 and next_row <= 8 and next_column >= 1 and next_column <= 8:
        result += 1
print(result)
```

이것이 취업을 위한 코딩 테스트다

1. 코딩 테스트 첫걸음

- 코딩 테스트 출제 경향 분석
- 알고리즘 성능 평가
- 파이썬 문법

이 자료는 동빈 나 님의 이코 테 유튜브 영상을 보고 정리한 자료입니다.

'Algorithm' 카테고리의 다른 글

[Algorithm] 16강 : 구현 유형 문제 풀이

[Algorithm] 14강 : 구현 유형 개요

[Algorithm] 13 강 : 그리디 유형 문제풀이 + 백준 알고리즘 11399번 ATM문제

[Algorithm] 12 강 : 그리디 알고리즘 개요(탐욕법)

[Algorithm] 11 강 : 자주 사용하는 라이브러리(유용한 라이브러리)

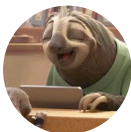
[Algorithm] 10 강 : 파이썬 문법 - 함수

algorithms 구현

알고리즘 구현

알고리즘 구현 유형

알고리즘 구현 유형 문제풀이



나무늘보스

혼자 끄적끄적하는 블로그 입니다.