

알튜비튜 백트래킹

오늘은 코딩테스트에 많이 나오는 알고리즘 중 하나인 백트래킹에 대해 배웁니다.
전번에 배운 완전탐색(브루트포스)에서 조금 더 발전해서, 더 이상 가망 없는 후보를 제외하고 탐색하는 알고리즘이죠.

백트래킹

- 완전탐색처럼 모든 경우를 탐색하나, 중간 과정에서 조건에 맞지 않는 케이스를 가지치기하여 탐색 시간을 줄이는 기법
- 모든 경우의 수를 탐색하지 않기 때문에 완전탐색보다 시간적으로 효율적임
- 탐색 중 조건에 맞지 않는 경우 이전 과정으로 돌아가야 하기 때문에, 재귀를 사용하는 경우 많음
- 조건을 어떻게 설정하고, 틀렸을 시 어떤 시점으로 돌아가야 할지 설계를 잘 하는 것이 중요

과정

- 어떤 노드의 유망성(promising)을 점검
 - 조건에 맞는지 안맞는지
 - 답이 될 수 있는지 없는지
- 유망하지 않다면(non-promising) 배제함 (가지치기)

가지치기

- 지금의 경로가 해가 될 것 같지 않으면(non-promising)
그 전으로 되돌아 가는 것(back)
- 즉, 불필요한 부분을 쳐내는 것
- 되돌아간 후 다시 다른 경로 검사
- 가지치기를 얼마나 잘하느냐에 따라 효율성이 결정됨



프로그래머스 : 소수 찾기 Lv.2

문제

- 0~9가 적힌 여러 개의 종이 조각이 존재
- 각 종이 조각에 적힌 숫자가 적힌 문자열 numbers가 주어질 때
- 조각들을 이어 붙여 만들 수 있는 소수의 개수를 구하는 문제

제한 사항

- 주어지는 조각의 개수는 1 이상 7 이하

예제 입력

"17"

예제 출력

3

접근

- N과 M 문제와 유사
→ 중복 X, 대신 길이 제한이 없다

접근

- 길이가 len 인 수열을 만드는 과정
: 길이가 1에서부터 점점 늘어난다
→ 과정 중에 길이가 $1, 2, \dots, (len-1)$ 인 수열이 만들어진다
- 최대 길이의 수열을 만들면서 중간 과정에서 만들어지는 수도 소수인지 체크
- 모든 길이의 수열을 다 고려하게 됨

/<> 2580번 : 스도쿠 - Gold 4

문제

- 빈칸이 있는 스도쿠 판이 주어짐
- 모든 빈 칸이 채워진 최종 모습을 출력하는 문제

제한 사항

- 채우는 방법이 여럿인 경우는 그 중 하나만을 출력

예제 입력

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 5 | 4 | 6 | 9 | 2 | 7 | 8 |
| 7 | 8 | 2 | 1 | 0 | 5 | 6 | 0 | 9 |
| 0 | 6 | 0 | 2 | 7 | 8 | 1 | 3 | 5 |
| 3 | 2 | 1 | 0 | 4 | 6 | 8 | 9 | 7 |
| 8 | 0 | 4 | 9 | 1 | 3 | 5 | 0 | 6 |
| 5 | 9 | 6 | 8 | 2 | 0 | 4 | 1 | 3 |
| 9 | 1 | 7 | 6 | 5 | 2 | 0 | 8 | 0 |
| 6 | 0 | 3 | 7 | 0 | 1 | 9 | 5 | 2 |
| 2 | 5 | 8 | 3 | 9 | 4 | 7 | 6 | 0 |

예제 출력

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 5 | 4 | 6 | 9 | 2 | 7 | 8 |
| 7 | 8 | 2 | 1 | 3 | 5 | 6 | 4 | 9 |
| 4 | 6 | 9 | 2 | 7 | 8 | 1 | 3 | 5 |
| 3 | 2 | 1 | 5 | 4 | 6 | 8 | 9 | 7 |
| 8 | 7 | 4 | 9 | 1 | 3 | 5 | 2 | 6 |
| 5 | 9 | 6 | 8 | 2 | 7 | 4 | 1 | 3 |
| 9 | 1 | 7 | 6 | 5 | 2 | 3 | 8 | 4 |
| 6 | 4 | 3 | 7 | 8 | 1 | 9 | 5 | 2 |
| 2 | 5 | 8 | 3 | 9 | 4 | 7 | 6 | 1 |

접근

- N-Queen 문제와 유사
- 재귀함수를 설계해보자
 - 모든 빈칸을 차례로 채워나감
 - 빈칸에 1~9까지 다 넣어봄
 - 가로, 세로, 정사각형에 중복되는 수가 있는 경우 가지치기

접근

- 가로 9칸, 세로 9칸, 정사각형 9칸 밖에 안되니까..
반복문으로 중복 체크
- O(N)

접근

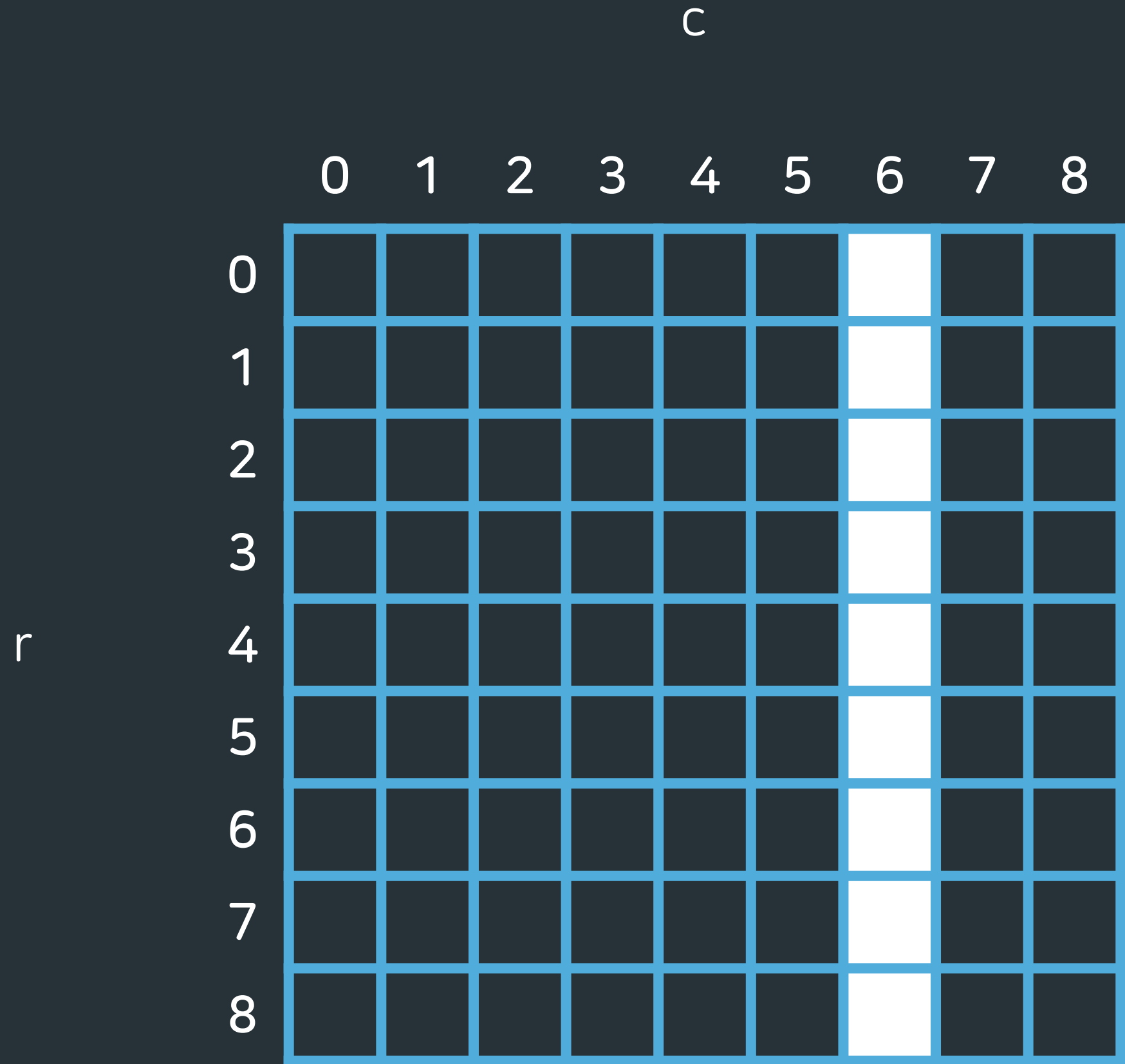
- N-Queen에서 했던 것처럼
가로, 세로, 정사각형 내 특정 숫자 존재 여부를 bool 배열로 관리
- O(1)
- 단, 퀸처럼 한 가지만 있는 것이 아니라 1~9까지 9종류 있으므로
2차원 bool 배열로 관리해야 함

C

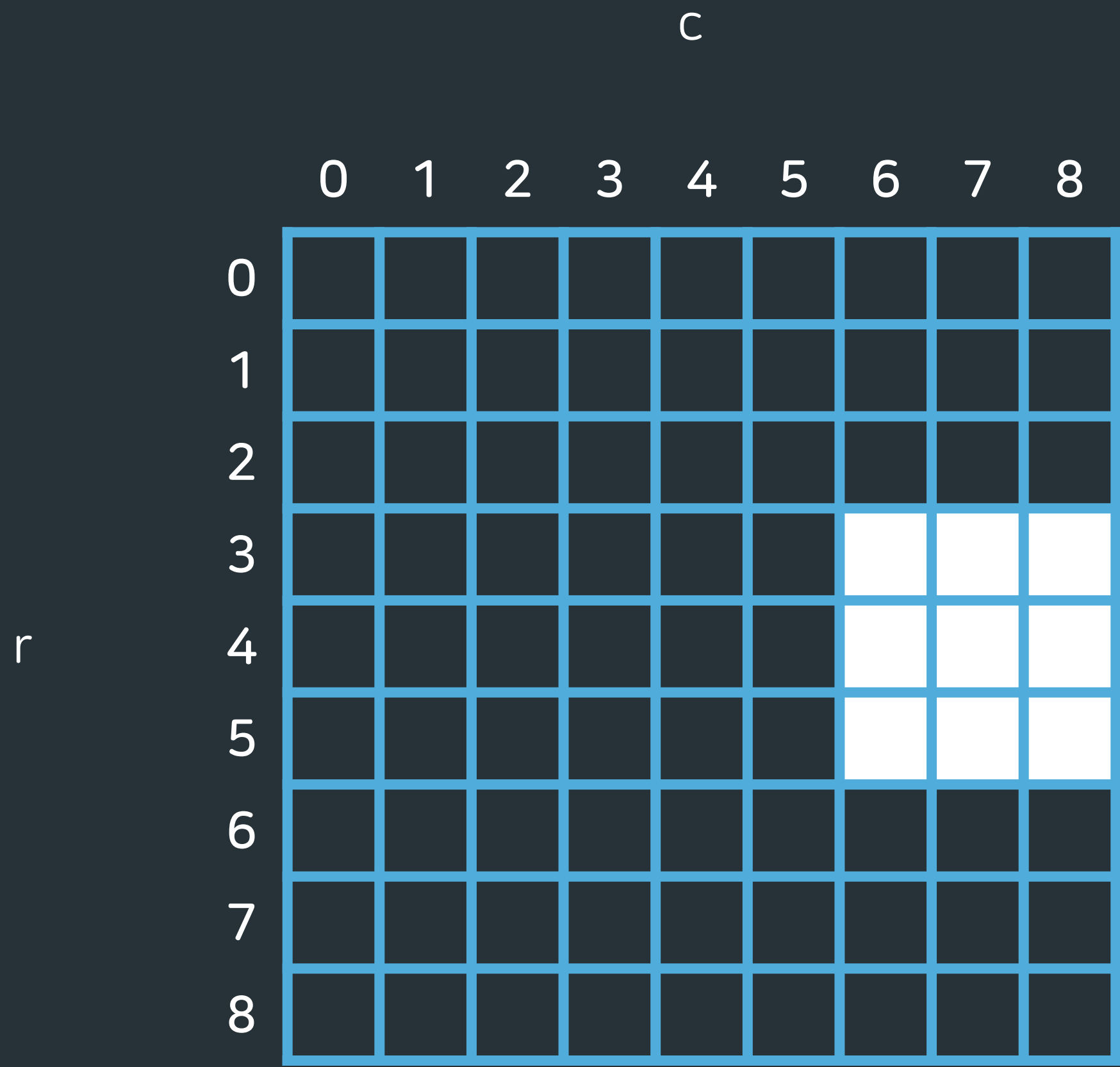
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | | |
| 1 | | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | | | | | | | | | |
| 4 | | | | | | | | | |
| 5 | | | | | | | | | |
| 6 | | | | | | | | | |
| 7 | | | | | | | | | |
| 8 | | | | | | | | | |

r

- `is_in_row[r][n]`
: r번 행에 n이 존재하는지 여부



- `is_in_col[c][n]`
: c번 열에 n이 존재하는지 여부



● is_in_square[s][n]
: s번 정사각형에
n이 존재하는지 여부

| | | $c / 3$ | | |
|---------|---|---------|---|---|
| | | 0 | 1 | 2 |
| $r / 3$ | 0 | 0 | 1 | 2 |
| | 1 | 3 | 4 | 5 |
| | 2 | 6 | 7 | 8 |

- $\text{square}(r, c)$
 $= (r / 3) * 3 + (c / 3)$

/<> 2477번 : 참외밭 - Silver 2

문제

- 1미터제곱의 넓이에 자라는 참외의 개수를 나타내는 양의 정수 K 가 주어진다.
- 육각형의 임의의 한 꼭짓점에서 출발하여 반시계 방향으로 돌레를 돌면서 변의 방향, 길이가 주어진다.
- 동쪽 1, 서쪽 2, 남쪽 3, 북쪽 4

제한 사항

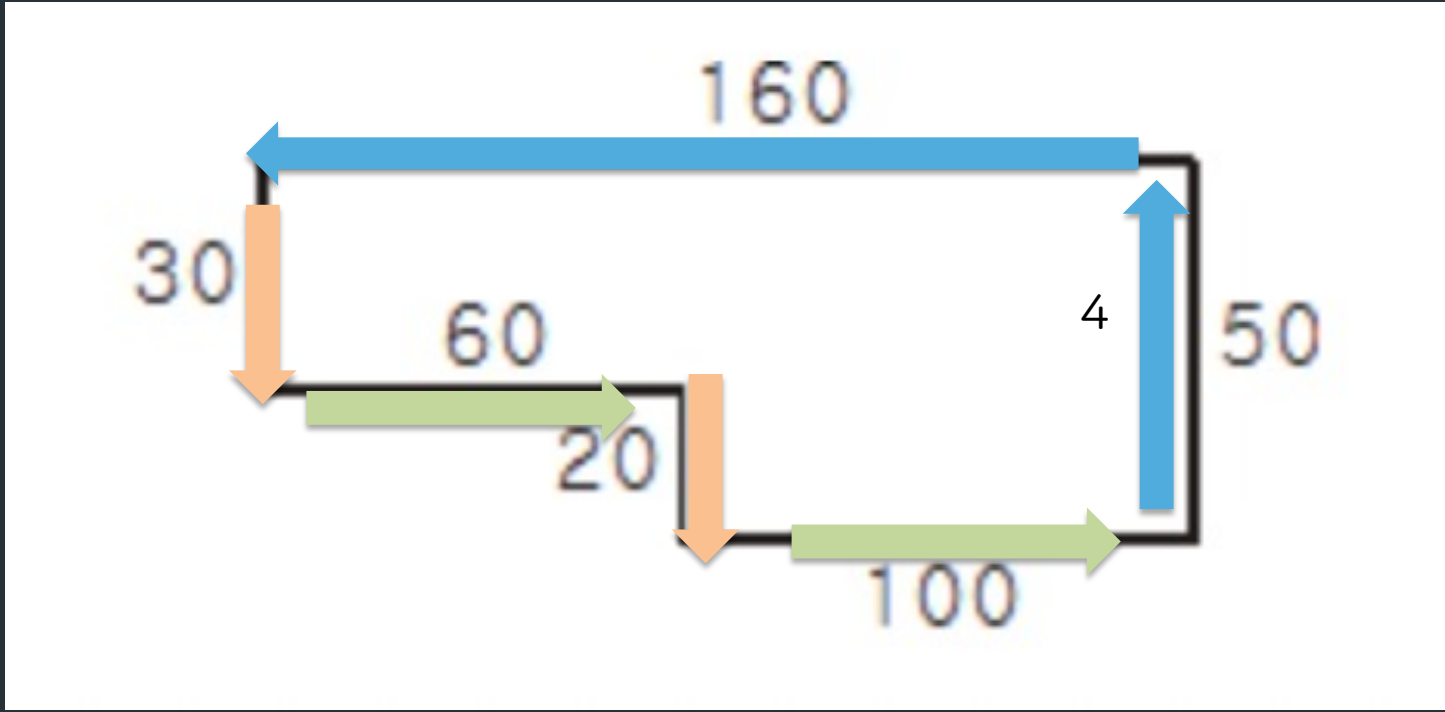
- $K(1 \leq K \leq 20)$

예제 입력

```
7
4 50
2 160
3 30
1 60
3 20
1 100
```

예제 출력

```
47600
```



접근

- 작은 사각형이 나타나는 조건을 생각해야함 → 같은 방향으로 나타나는게 한 칸 건너서 나타날 때
- 큰 사각형 : 같은 방향으로 나타나는 세로의 합 * 같은 방향으로 나타나는 가로의 합
- 임의의 한 꼭짓점에서 출발한다는 점을 고려

```
int findArea() { // 육각형 모양의 영역 너비를 구하는 함수
    int big_square;
    int small_square;
    for (int i = 0; i < 9; i++) {
        if (li[i].first == li[i + 2].first && li[i + 1].first == li[i + 3].first) { // 같은 방향이 한 칸을
            건너뛰고 나타나는 형태이면
                big_square = (li[i].second + li[i + 2].second) * (li[i + 1].second + li[i + 3].second); //
                전체 사각형 갱신
                small_square = li[i + 2].second * li[i + 1].second; // 작은 사각형(영역이 없는 구역 갱신)
            }
        }
    }
    return big_square - small_square;
}
```

추가로 풀어보면 좋은 문제!

- /<> 15657번 : N과 M (8) – Silver 3
- /<> 10971번 : 외판원 순회 2 – Silver 2
- /<> 14889번: 스타트와 링크 – Silver 2
- /<> 13023번: ABCDE – Gold 5
- /<> 1759번 : 암호 만들기 – Gold 5
- /<> 15811번 : 복면산?! – Gold 4