

Daily Contents

SW 역량 테스트

김사덕 실습코치

역량 테스트란?

- 삼성전자에서 주관하는 SW 검정시험
- A, A+, B, C 총 4가지 등급을 가지고 있으며 등급에 따라 우대하고 있음
- 공채 역량 테스트, 상시 역량 테스트 2가지가 있음
- 삼성스러운 SW GSAT -> 복잡한 상황을 이해할 수 있는가
- 시뮬레이션, DFS, BFS 3개의 카테고리에 대부분 포함

얼마나 우대?

- A : 서류 전형 우대
- A+ : 서류 전형 우대, 역량 테스트 우대
- B : 서류 전형 우대, 역량 테스트 우대, 면접 전형 우대

단, 2년간 유효

SW 역량 테스트

역대 공채 SW 역량 테스트 기준

연도	브루트 포스	BFS	시뮬레이션 및 구현	기타
2015 하	1	0	0	1
2016 하	1	0	1	0
2017 상	3	0	1	0
2017 하	4	0	0	0
2018 상	3	0	1	0
2018 하	0	1	4	0
2019 상	1	1	3	0
2019 하	2	1	2	0
2020 상	1	1	3	0

최근 난이도 높은 문제

백준 17387 - 새로운 게임 2

백준 19236 - 모노미노도미노

배열에서 객체들을 잘 컨트롤 할 수 있는가

Question

질문 있어요!

Question

Question

1. DFS, BFS 따지만 공부하면 되나요?

No! ...?

공채 역량 테스트에서는 그 이상이 나온 적 없음

But 작년 9월 상시 역량 테스트에서 MST가 나온 적 있음

즉, Prim, Kruskal, Dijkstra 관련 문제가 나올 수도 있음

자신의 현 상황에 맞게 작전을 짜는 것이 중요합니다.

Question

2. 예시 테스트 케이스만 통과하면 끝인가요?

No! IM 시험과 다릅니다.

A형에서는 10개의 테스트 케이스를 제공합니다.

하지만 10개의 테스트 케이스를 맞았다고 해서 정답이 아닙니다.

**SW Expert Academy에서 모의 SW 역량문제를 풀어보시면
시험에 대한 감을 잡으실 수 있습니다.**

Question

3. A형도 역량 테스트 우대를 받을 수 있나요?

SSAFY에서만 A형만 가지고 있는 교육생에게 공채를 지원하기 전, 시험을 칠 것인지 A형을 시험 결과로 넣고 시험을 치지 않을 것인지를 물어봅니다. 하지만 그렇게 시험을 치지 않은 A형이 면접 전형으로 갈 수 있다는 보장은 없습니다.

Question

4. 공부할 곳 추천해주세요!

삼성 SW 역량 테스트에 맞는 곳만 알려드리겠습니다.
Mattermost에 올려진 글을 확인해주세요.

네카라는 시험 유형이 다르기 때문에 코딩 테스트 준비를
다르게 진행하셔야 합니다. Ex) 프로그래머스

Question

Question

5. 백준에 있는 기출이 실제 문제랑 동일한가요?

No!

시험 응시자들이 문제를 풀고 나서 문제에 대해
'백준'님에게 문제에 대해서 간략히 전달하면 '백준'님이
문제를 만드시는 형식이기 때문에 실전과 조금 다릅니다.

Tip

- 대부분의 시험이 1번, 2번 난이도가 비슷하거나 1번이 더 쉬운 편입니다. 그러니 1번부터 푸는 것이 현명!
- 시험치기 전, 자신이 1솔이 목표인지 2솔이 목표인지를 정하는 것이 좋습니다.
- 문제를 많이 푸시면서 알고리즘 테크닉을 만드셔야 합니다. 순열, 조합 등 기본적인 알고리즘은 필수!
- 백준 사이트를 이용하실 때, `solved.ac`를 사용하시면 난이도에 대한 정보를 확인하실 수 있습니다.

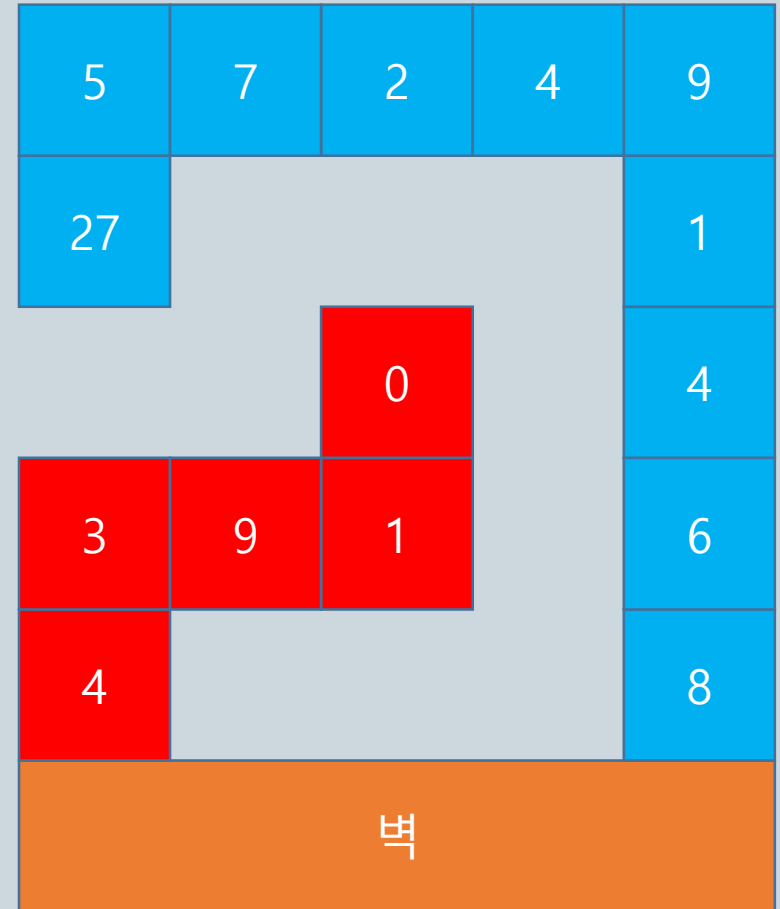
풀이 Tip!

방향 테크닉

- $dy = \{0, 1, 0, -1\};$
 $dx = \{1, 0, -1, 0\};$
- i 번째 인덱스일 때,
for문에서 시계 방향으로 움직일 때, $i = (i+1)\%4$
for문에서 반시계 방향으로 움직일 때, $i = (i+3)\%4$

배열 테크닉

- 연결된 곳들의 합을 구할 때,
벽에 붙어있으면 하나의 도형으로
취급한다면 연결성을
어떻게 처리하는 것이 좋을까?



배열 테크닉

- **방법 1**
5x5 행렬을 만들어 가장 아래 부분에서 for문을 통한 dfs
- **방법 2**
벽을 포함한 6x5 행렬을 만들어 (0,0)에서 한번만 dfs

5	7	2	4	9
27				1
				4
3	9	1		6
4				8
벽				

배열 테크닉

요약

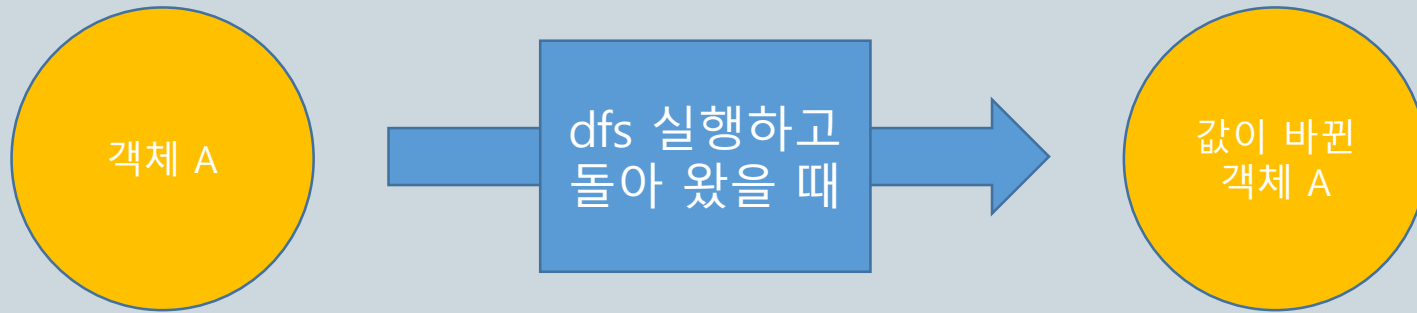
기존 배열보다 size를 1 혹은 2 크게 만들어서 코드를 작성하게 되면 간단하게 바뀌는 경우가 많습니다.

dfs 테크닉

- dfs 재귀를 사용할 경우
아래 depth의 과정을 모두 수행하고 돌아왔을 때,
값을 되돌려야 합니다.
하지만 그 상황이 까다롭다면?

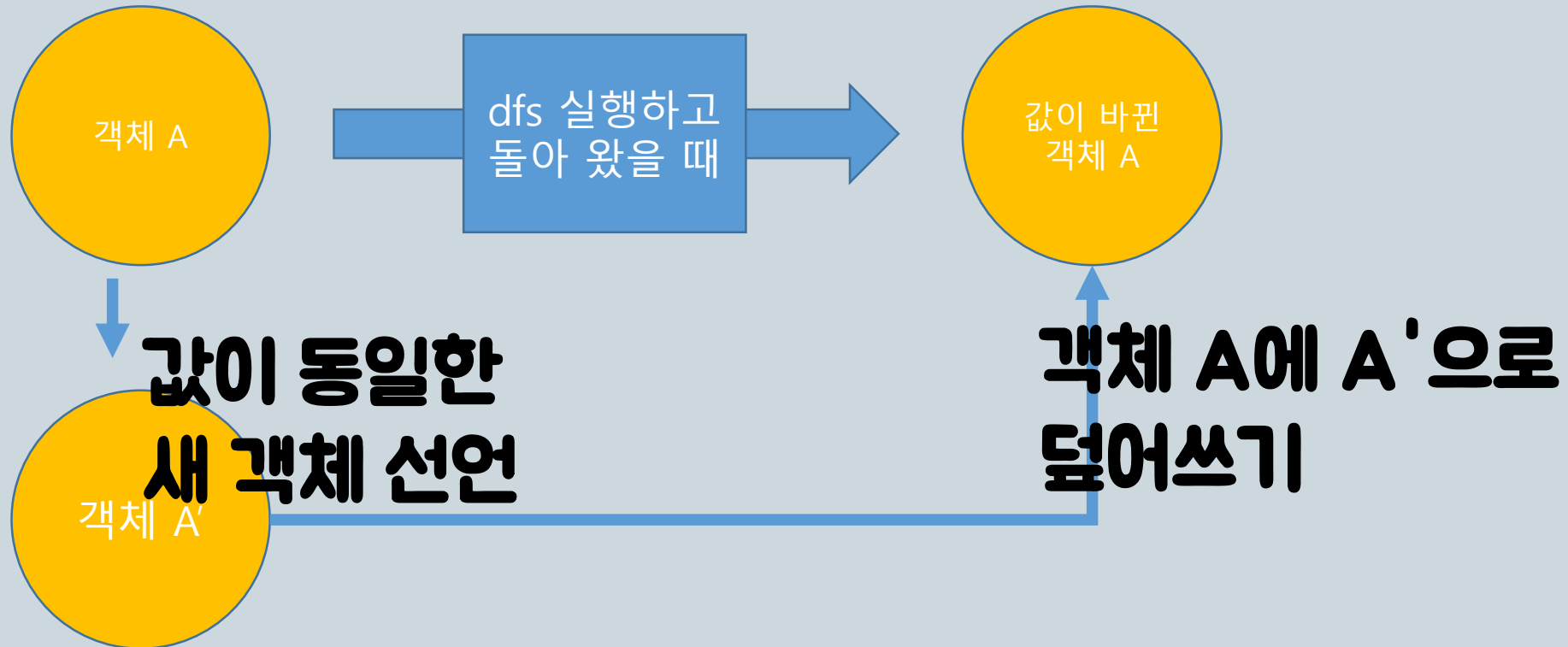
Ex) 객체들이 서로 영향을 줘서 이전 상태로
되돌리는게 까다로운 상황

dfs 테크닉 - 값 덮어쓰기

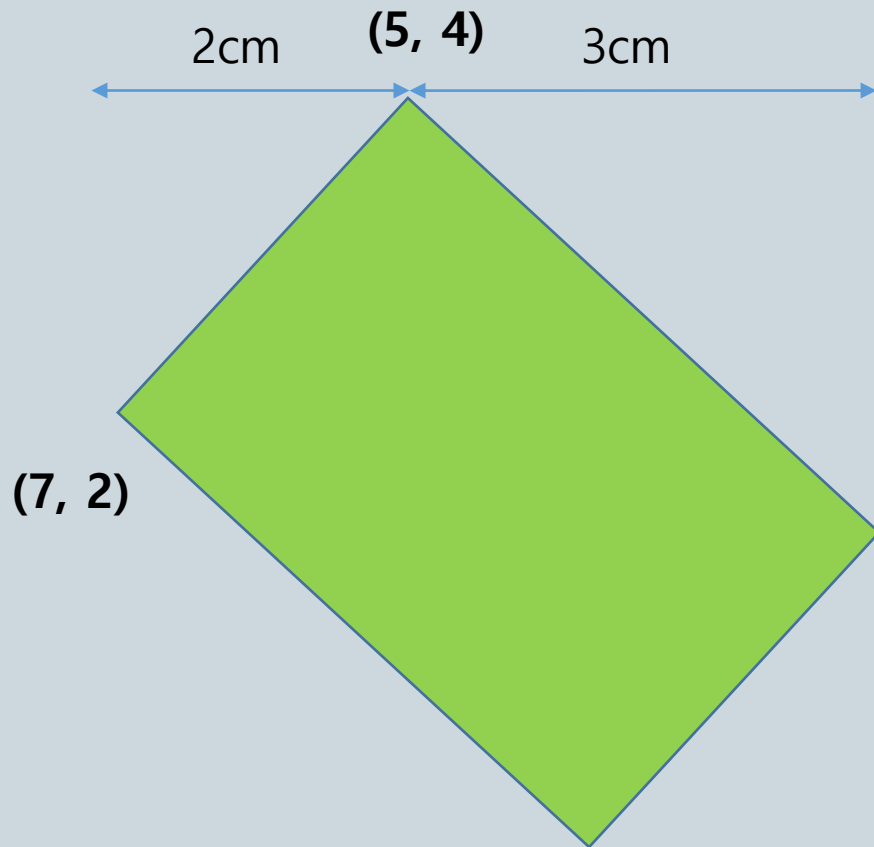


**어떻게 이전의 객체
A의 상태로 돌릴 수 있을까?**

dfs 테크닉 - 값 덮어쓰기



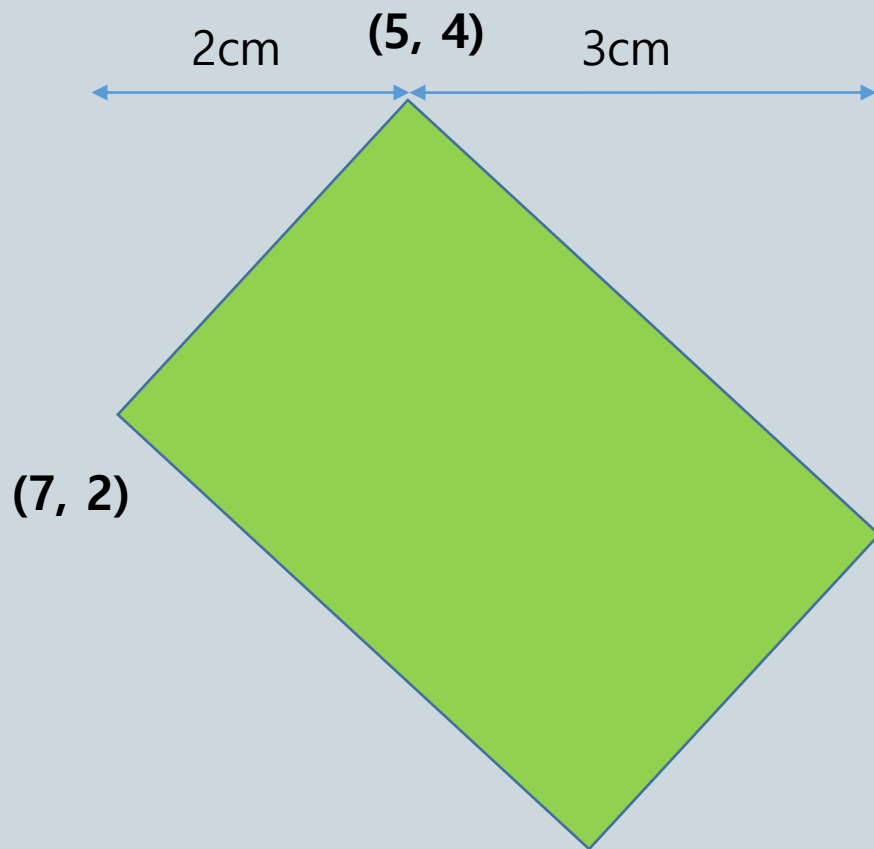
대각선 테크닉



- 배열에서 대각선 테두리 및 안에 있는 값들의 합 구하기
- 배열 이름을 arr라고 하고 임의의 값들이 다 들어있음

알고리즘 테크닉

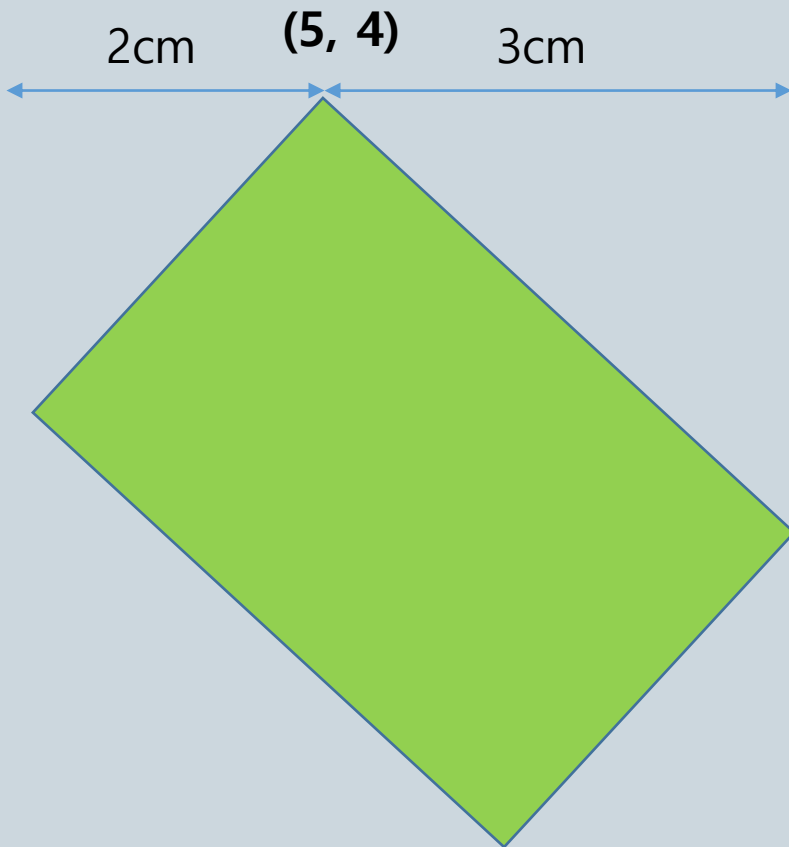
Java



```
int sum = 0, y = 5, x = 4;
for (int i=0; i<=2; i++){
    for(int j=0; j<=3; j++){
        sum += arr[y+i+j][x-i+j];
    }
}

for (int i=0; i<2; i++){
    for(int j=0; j<3; j++){
        sum += arr[(y+1)+i+j][x-i+j];
    }
}
```

Python



```
sum = 0, y = 5, x = 4;
for i in range(3):
    for j in range(2):
        sum += arr[y+i+j][x-i+j];
for i in range(2):
    for j in range(1):
        sum += arr[(y+1)+i+j][x-i+j];
```

SW 역량 테스트

감사합니다!