

lambda 표현식

```
1. def discriminator(a):  
    if a == 1:  
        return a == 1
```

맞는 말.

2. lambda 표현식 → inline 함수 → 호출 X → 속도 ↑

C/C++ inline
컴파일러

비트
전처리 → 컴파일 → Obj → 실행

X

→ lambda 표현식도 함수 호출해서 사용
⇒ 일회성 함수 성능 =.

Sorting rule
↑

VS
≠

lambda

가득 ↑

재귀.

- ① Base Case \Rightarrow 밑바닥 케이스?
기저 사례.
- ② Recursive case. \Rightarrow 반복적으로 재귀

def factorial (n):
 if n == 1:
 return 1
 else:
 return n * factorial (n-1).

$n=3$

Base case

recursive case

$3 \times 2 \times 1$

3 10

.....

...XX...XX...

XXX.....

1. 성이 가라앉나 인 가라앉나 ✓

⇒ 이중 반복문.

2. Row, Col 을 줄여야 함

⇒ 이중 반복문

Representation
변환.

[10]

[2, 0, 0, 1, 0, 0, 1]

[0, 0, 0, 7]

.....

..XX...X..

..XX.....

[2, 3, 3, 2]

[1, 1, 1, 1]

1 2 3 4 5

[2, 0, 3, 2]

[~~CX~~ [2, 3, 3, 2]]

성민개수 $\frac{n}{2}$

$$O(rc) \rightarrow 25$$

$$O(n \log n)$$

$$O(\text{성민개수} \log (\text{성민개수}))$$

$$\underline{r=5} \quad \underline{c=5}$$

$$n=20$$

$$20 \log 20 \approx 44$$

90