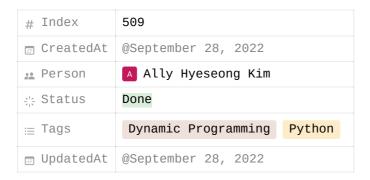
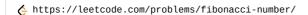
Fibonacci Number



References

LeetCode - The World's Leading Online Programming Learning Platform Level up your coding skills and quickly land a job. This is the best place to expand your knowledge and get prepared for your next interview.





파이썬 알고리즘 인터뷰

2021 세종도서 학술부문 선정작. 현업과 실무에 유용한 주요 알고리즘 이론을 깊숙이 이 해하고, 파이썬의 핵심 기능과 문법까지 상세하게 이해할 수 있는 취업용 코딩 테스트를 위한 완벽 가이드다. 200여 개가 넘는...

ttps://www.aladin.co.kr/shop/wproduct.aspx?ItemId=245495826



References

- 1. Bottom-Up Dynamic Programming
- 2. Brute Force
- 3. Top-Down Dynamic Programming
- 4. Numpy

1. Bottom-Up Dynamic Programming

1.1. Solution 1

```
class Solution:
    def __init__(self):
        self.f = [0, 1]

    def fib(self, n: int) -> int:
        for _ in range(2, n + 1):
            self.f.append(0)

    if n == 0 or n == 1:
        return self.f[n]

    for i in range(2, n + 1):
        self.f[i] = self.f[i - 1] + self.f[i - 2]

    return self.f[n]
```

1.2. Solution 2

Fibonacci Number 1

```
class Solution:
    dp = collections.defaultdict(int)

def fib(self, n: int) -> int:
    self.dp[0] = 0
    self.dp[1] = 1

for i in range(2, n + 1):
    self.dp[i] = self.dp[i - 1] + self.dp[i - 2]

return self.dp[n]
```

1.3. Solution 3

```
class Solution:
    def fib(self, n: int) -> int:
        x, y = 0, 1

    for i in range(0, n):
        x, y = y, x + y

return x
```

2. Brute Force

```
class Solution:
    def fib(self, n: int) -> int:
        if n <= 1:
            return n

    return self.fib(n - 1) + self.fib(n - 2)</pre>
```

3. Top-Down Dynamic Programming

```
class Solution:
    dp = collections.defaultdict(int)

def fib(self, n: int) -> int:
    if n <= 1:
        return n

    if self.dp[n]:
        return self.dp[n]

self.dp[n] = self.fib(n - 1) + self.fib(n - 2)

return self.dp[n]</pre>
```

• collections.defaultdict(default): dictionary value 가 주어지지 않을 경우, default 자료형으로 초기화한다.

```
    collections.defaultdict(int): 0으로 초기화
    collections.defaultdict(list): list 로 초기화
    collections.defaultdict(set): set 으로 초기화
```

4. Numpy

Fibonacci Number 2

$$\begin{bmatrix} F_{i+1} \\ F_i \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_i \\ F_{i-1} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^i \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^i \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$(2)$$

$$(3)$$

$$=\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}^i \begin{bmatrix}F_1\\F_0\end{bmatrix} \tag{2}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^i \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3}$$

```
class Solution:
    def fib(self, n: int) -> int:
    m = np.matrix([[0, 1], [1, 1]])
         vec = np.array([[0], [1]])
          return np.matmul(m ** n, vec)[0]
```

Fibonacci Number 3