피보나치 수를 구하라.

Input: n = 2
Output: 1
Explanation: F(2) = F(1) + F(0) = 1 + 0 = 1.

1.브루트 포스 재귀

```
class Solution:
    def fib(self, n: int) -> int:
        if n <= 1:
            return n
        return self.fib(n - 2) + self.fib(n - 1)
```

2.메모이제이션(memoization)

```
import collections

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

- 하향식

3.타뷸레이션(tabulation)

```
import collections

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 - 2] + self.dp[i - 1]
        return self.dp[n]
```

- 상향식

4.두개 변수만 사용

```
class Solution:
    def fib(self, n: int) -> int:
        x, y  = 0, 1
        for i in range(n):
            x, y = y, x + y
        return x
```

5.행렬

```
import numpy as np

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]
```

- O(logn)으로 행렬의 n승을 계산하는 방식