거듭제곱을 이용한 피보나치 수 구하기

태그	알고리즘
세부 주제	알고리즘 파이썬
검색어	피보나치 행렬곱
URL	

우선 풀려고 했던 문제는 백준의 `11444번 문제 - 피보나치 수 6` 난도는 `골드 2`

11444번: 피보나치 수 6
피보나치 수는 0과 1로 시작한다. 0번째 피보나치 수는 0이고, 1번째 피보나치 수는 1이다. 그 다음 2번째 부터는 바로 앞 두 피보나치 수의 합이 된다.
https://www.acmicpc.net/problem/11444

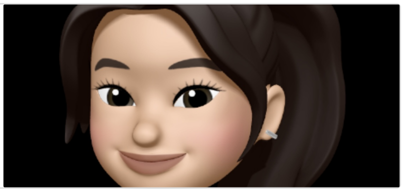
BAEKJOON ONLINE JUDGE

대충 풀던 방식으로 구현해서 돌리다가 컴퓨터를 한번 프리징 시키고 나서 풀이 방법을 찾아보았다.

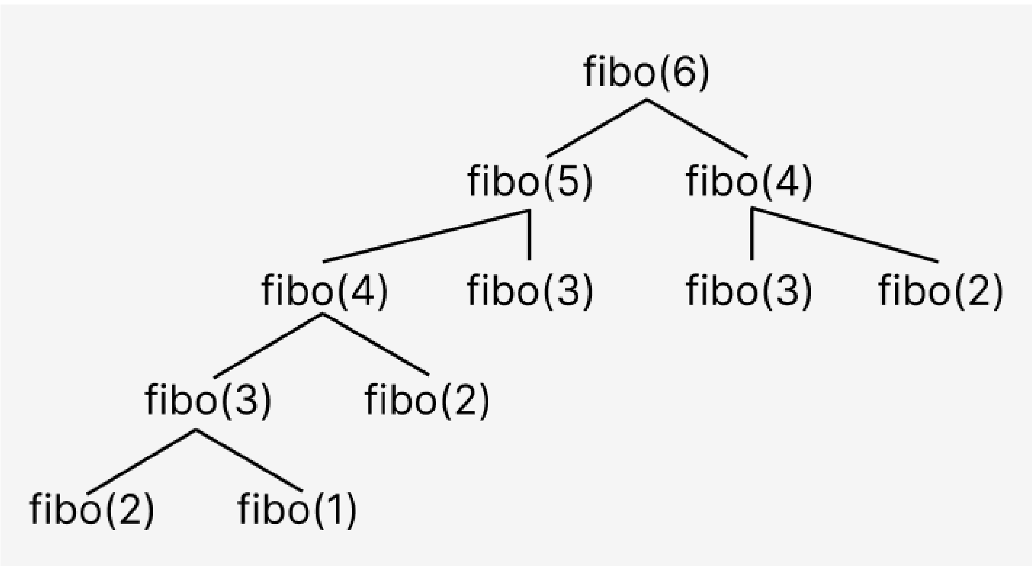
찾아 보니 행렬곱으로 구하면 된다는 말이 나왔다.

이 내용은 이 블로그 참조

백준 11444 피보나치 수 6 (파이썬)
11444 피보나치 수 6 바로가기
https://ataraxiady.github.io/dev/2021/04/15/dev-boj-2_11444/

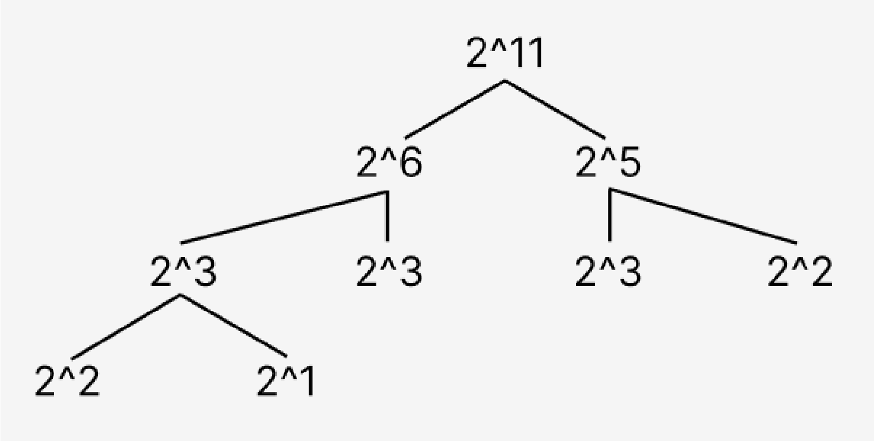


우선 구하던 방식으로 구하다보면 5번째 피보나치 수를 구하기 위해 무조건 1부터 4까지의 피보나치 수를 알아야 하고 이는 dp를 이용하더라도 n번 연산을 해야한다.



그래서 주어진 n의 조건에 따라 연산을 1,000,000,000,000,000,000(백 경) 회 해야 한다는 의미이다.

그런데 여기서 거듭제곱을 이용하게 되면 이 연산 횟수가 $O(logn)$ 수준으로 줄어들게 된다



숫자가 n이 11임에도 불구하고 위의 fibo(6)보다 트리의 높이가 낮은 것을 확인할 수 있다.

그러면 다시 돌아와서, 피보나치 수를 거듭제곱으로 나타내는 방법에는 행렬곱이 있다고 한다.

이 내용은 위의 블로그에서도 링크를 걸어두신 다음 백준님의 블로그를 보면 나와있다.

피보나치 수를 구하는 여러가지 방법

2749번 문제: 피보나치 수 3번은 지금까지 풀었던 세 문제와 똑같이 N번째 피보나치 수를 구하는 문제입니다. 그런데, N이 10 18 로 매우 큽니다. 다행히도 1,000,000로 나눈 나머지를 출력하는 문제네요.

https://www.acmicpc.net/blog/view/28

그런데 설명이 간략하게 작성되어있다보니 따라가다 막히는 부분이 있었고 중간 부분을 채워보기로 한다.

$$\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}$$

위의 식을 정리하면 다음과 같이 나온다고 하는데, 이를 설명해보자면

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n$$

일단 위의 식 중 첫번째 식을 다음과 같이 쓸 수 있다.

$$\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix}$$

이를 반복해서 표현하다 보면

$$\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} F_2 \\ F_1 \end{pmatrix}$$

이 되고,

n을 한단계만 앞당겨서 써보면

$$\begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} F_2 \\ F_1 \end{pmatrix}$$

이 된다.

그리고 두번째 식은 바로 위의 식을 합쳐서 표현할 수 있는데

$$\begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} F_2 \\ F_1 \end{pmatrix}, \begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} F_1 \\ F_0 \end{pmatrix}$$

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} F_2 & F_1 \\ F_1 & F_0 \end{pmatrix}$$

위와 같이 표현할 수 있고 $F_2 = F_1 = 1, F_0 = 0$ 이므로, 백준님이 정리한 두번째 식이 나온다.

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n}$$

이것을 행렬곱을 구현해서 풀이해도 되지만, 행렬곱을 구현하면 코드를 이해하는데 불편한 것 같아서 이를 덧셈식으로 대체해 보려고 한다.

우리가 궁극적으로 구해야하는 것은 F_n이므로 여기에 대해서만 생각해보려고 한다.

일단 거듭제곱을 $O(logn)$의 시간복잡도로 연산하는 방법은 주어진 지수 n이 짝수일 때와 홀수 일 때로 나뉘므로 마찬가지로 짝수와 홀수로 나누어 생각해보도록 하겠다.

짝수일 때

$n = 2m$으로 두고 위의 마지막 식을 변형해보면

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{m} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{m}$$

이고, $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{m}$ 을 F_m에 관련된 식으로 바꾸어 정리하면

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} F_{m+1} & F_m \\ F_m & F_{m-1} \end{pmatrix} \begin{pmatrix} F_{m+1} & F_m \\ F_m & F_{m-1} \end{pmatrix}$$

이 된다. 여기서 우리가 원하는 F_n만 표현하자면,

$$F_n = F_{m+1}F_m + F_mF_{m-1}$$

로 나타낼 수 있다. 이 때 F_{m+1}과 F_m, F_{m-1}로 3개의 미지수가 생기게 되어 3번의 함수 호출(혹은 참조)가 필요하게 되는데 우리는 이미 F_{m+1}을 알고 있다. 바로 $F_{m+1} = F_m + F_{m-1}$!! 이 식을 적용하면

$$F_n = (F_m + F_{m-1})F_m + F_mF_{m-1}$$

이 되고 이 식을 사용하면 된다.

홀수일 때

$n = 2m + 1$로 두고 위의 마지막 식을 변형해 보면

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{m} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{m+1}$$

이고, $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{m}$ 과 $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{m+1}$ 을 F_m에 관련된 식으로 바꾸어 정리하면

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} F_{m+1} & F_m \\ F_m & F_{m-1} \end{pmatrix} \begin{pmatrix} F_{m+2} & F_{m+1} \\ F_{m+1} & F_m \end{pmatrix}$$

이 된다. 여기서 우리가 원하는 F_n만 표현하자면,

$$F_n = {F_{m+1}}^2 + {F_m}^2$$

이 되고 이 식을 사용하면 된다.

위의 식을 이용하여 피보나치를 구현한 파이썬 코드

```
def fibo(n: int) -> int:
    if n == 0:
        return 0
    if n < 3:
        return 1
    elif n % 2 == 0:
        fm = fibo(n // 2)
        fm_1 = fibo(n // 2 -1)
        return (fm + fm_1) * fm + fm * fm_1
    else:
        return fibo(n // 2 + 1) ** 2 + fibo(n // 2) ** 2

n = int(input())
print(fibo(n))
```

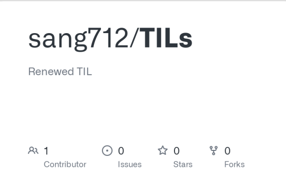
여기에 문제에서 준 조건인 나머지를 구하는 조건과, 시간 단축을 위한 메모이제이션을 추가하면 된다.

백준 제출 코드

TILs/BOJ11444_피보나치 수 6.py at 077763c8c1617349d1541bddff90907e613bcb60 · sang712/TILs

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

https://github.com/sang712/TILs/blob/077763c8c1617349d1541bddff90907e613bcb60/Algorithm/BOJ11444_%ED%94%BC%EB%B3%B4%EB%82%98%EC%B9%98%20%EC%88%98%206.py



참고를 한 블로그

[BOJ] 11444. 피보나치 수 6

백준 문제 주소 : https://www.acmicpc.net/problem/11444 11444번: 피보나치 수 6 첫째 줄에 n이 주어진다. n은 1,000,000,000,000,000,000보다 작거나 같은 자연수이다. www.acmicpc.net 알고리즘은 수학과 밀접한 관련이 있다는 걸 깨닫게 해주는 문제였습니다... (불태웠어,,,,) 피보나치 수는 재귀로 쉽게 풀 수 있

https://txegg.tistory.com/109