[자료구조 및 알고리즘 과제]

[스마트 ICT 융합공학과 201811567_주용한]

[피보나치 수열의 시간 복잡도]

pib(n)을 피보나치 함수라고 하자.

입력 받은 n 값이 0, 1 이 될 때까지 return pib(n-1)+pib(n-2); 을 실행한다.

따라서 pib(n)의 연산 시간은

T(0) = T(1) = c(상수),

n>2 일때, T(n) = T(n-1) + T(n-2) + c(상수) 로 볼 수 있다.

위의 식을 통해 T(n) > T(n-1) 부등식이 성립하고

따라서 T(n) > 2 * T(n-2)

>4 * T(n-4)

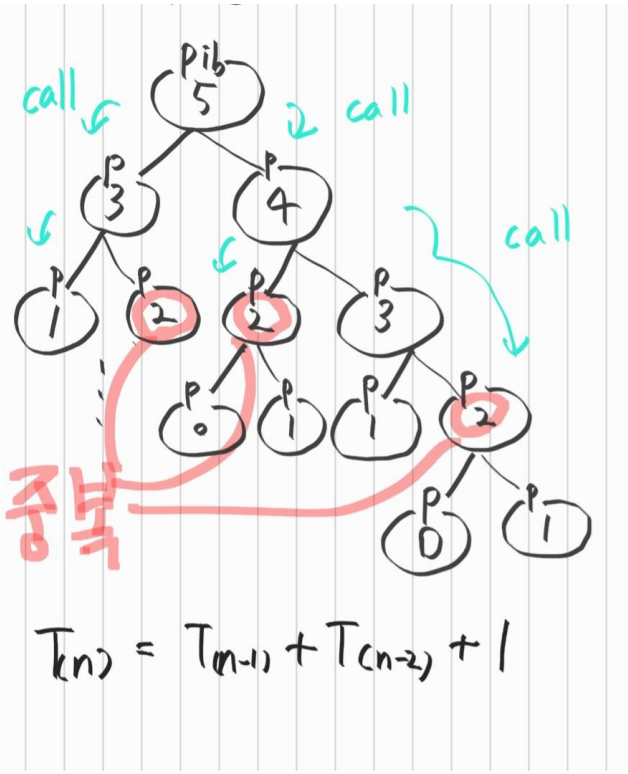
>8 * T(n-6)

…

>2^n/2 * T(0) = c*2^n/2

시간 복잡도는 O(2^n)이 된다.

[피보나치 수열의 문제점]



pib(n) 을 구하기 위해 pib(n-1),pib(n-2) 2 개의 함수를 다시 호출 해야 하므로 재귀호출을 한번 실행할 때 마다 호출함수의 수는 2 배로 증가한다.

이는 매우 비효율 적이며 많은 중복을 포함한다.

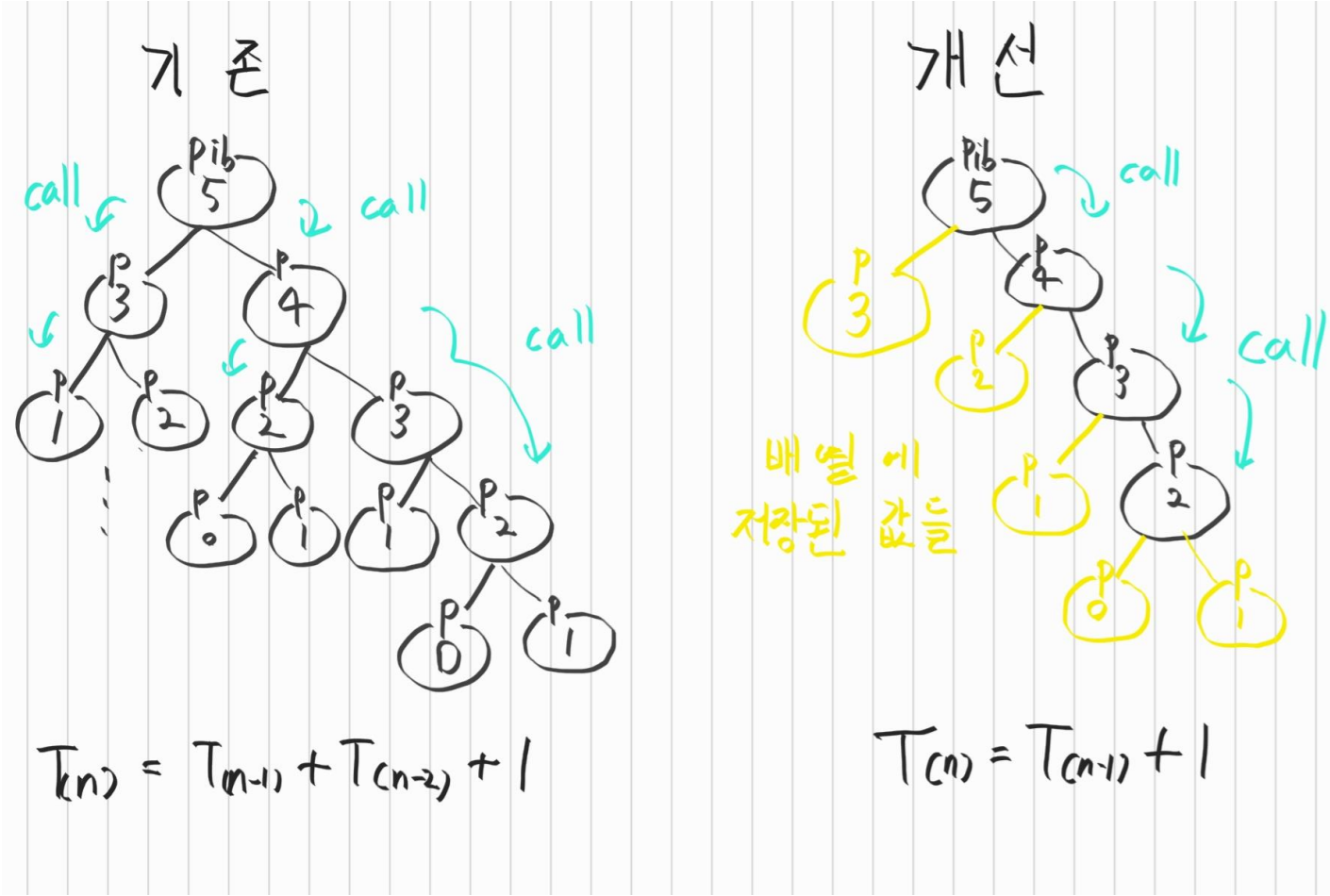
왼쪽 그림을 보면 알 수 있듯 이미 계산한 값을 중복해서 호출을 하고 있다.

(예를 들어 pib(5)를 호출하기 위해서는 pib(2)이 3 번이나 호출된다.)

[문제점 개선하기]

pib(5)에서부터 양쪽으로 뻗어가는 트리 모양을 피하고 한쪽으로만 뻗어 나가는(하나의 재귀함수만을 호출) 함수를 생각해 보았다.

그리고 중복한 값을 계속 계산하지 않기 위해 이미 계산한 값을 배열에 넣고 다음에 다시 사용하기로 했다.



위 그림과 같이 pin(n) = pib(n-1) + A(배열에 저장된 수)가 된다면

시간복잡도는 T(n) = T(n-1) + c

= T(n-2) + 2c

= T(n-3) + 3c

…

=T(0) + n*c

실제 코드 수정내용 중 일부분

return pib(num-1) + pib(num-2);

↓

return *(arr+num) = improvedPib(num-1, arr) + *(arr+num-2);

이므로 O(n)의 시간복잡도를 가지게 되고 문제점을 해결 할 수 있다.