

# 스터디 3주차

재귀,스택,큐

# 재귀함수

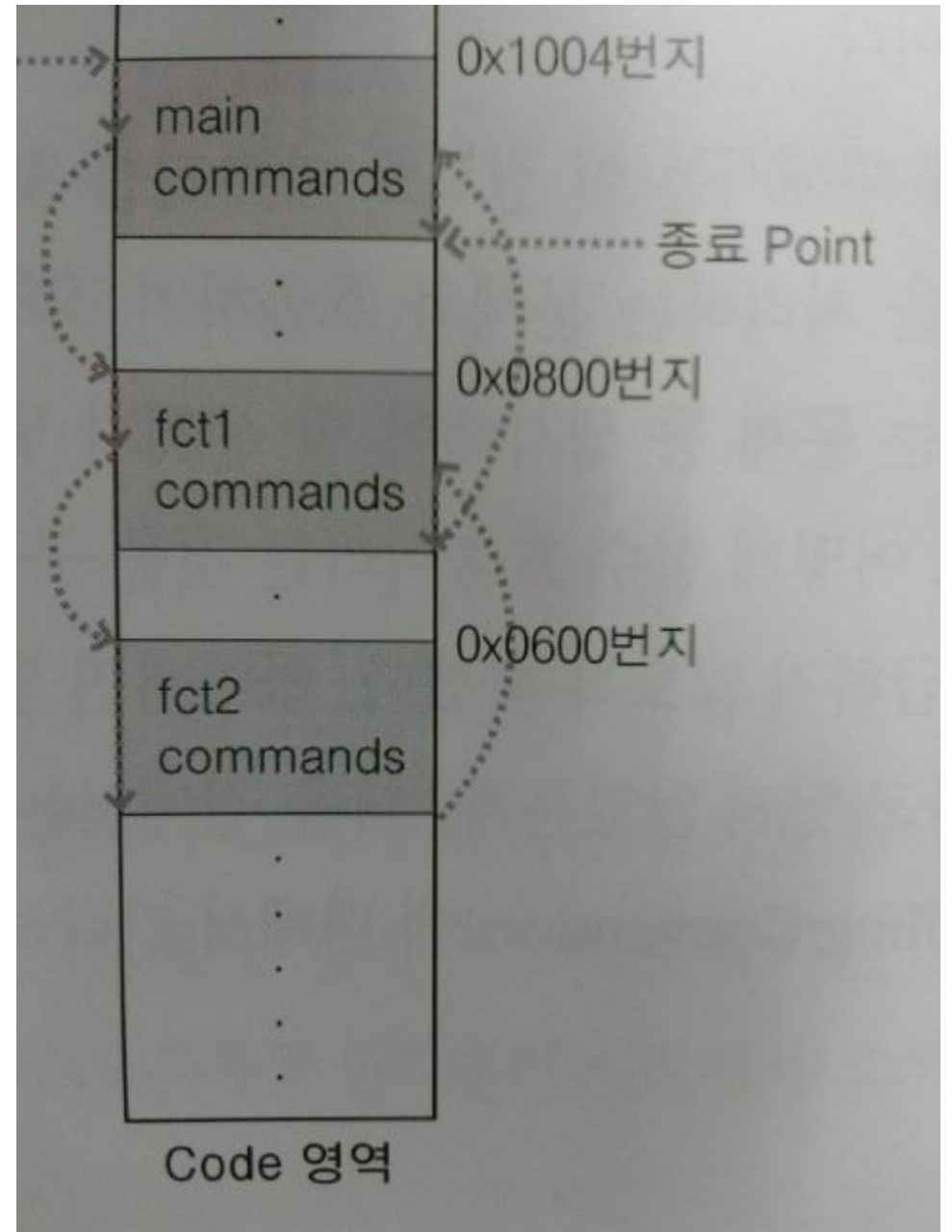
- 함수 안에서 본인 함수를 호출하는 방식을 'Recursion' 또는 재귀함수 라고 한다.
- 필자는 고등학교~19년 4월까지 재귀를 쓰는 사람들을 '번태'라고 불렀다.

# 재귀함수 왜 쓸까?

- 알고리즘 자체가 재귀적으로 표현하기 자연스러울 때
  - 흔히 점화 식으로 표현할 수 있으면 재귀가 정말 좋다.
- 변수 사용을 줄여준다.
  - 메모리 공간을 잡아먹는다는 의미가 아니라, 'mutable state' 즉 변경 가능한 상태를 제거하여 프로그램 오류가 발생할 수 있는 가능성을 제거 한다는 이야기 이다.
- 당장에는 stack을 위해. 더 나아가 DFS를 배우기 위해.
  - DFS는 가중치가 없을 때 최단경로 알고리즘이다.

# 재귀

- 오른쪽 방식으로 함수가 호출되면 스택에 쌓인다.
- 재귀는 본인 함수를 계속 호출하니, 계속 쌓인다.
- 일정 이상 쌓이면 메모리 오류가 나서 프로그램이 터진다.
- 그러니 **종료조건**을 잘 설정해야 한다.



# 재귀

- 우리가 시험기간때 꼭 봤던 피보나치를 보자.

```
int Fibo(int n) {  
    printf("현재 호출된 함수 %d\n", n);  
    if (n == 1) return 0;  
    else if (n == 2) return 1;  
    else return fibo(n - 1) + fibo(n - 2);  
}
```

Fibo(5)
Fibo(6)
Fibo(7)

함수 호출을 하면 오른쪽 그림과 같이 스택에 쌓인다고 한다.  
Fibo(5)는 return 부분 왼쪽일까? 오른쪽일까?

# 재귀

- 우리가 시험기간때 꼭 봤던 피보나치를 보자.

```
int Fibo(int n) {  
    printf("현재 호출된 함수 %d\n", n);  
    if (n == 1) return 0;  
    else if (n == 2) return 1;  
    else return fibo(n - 1) + fibo(n - 2);  
}
```

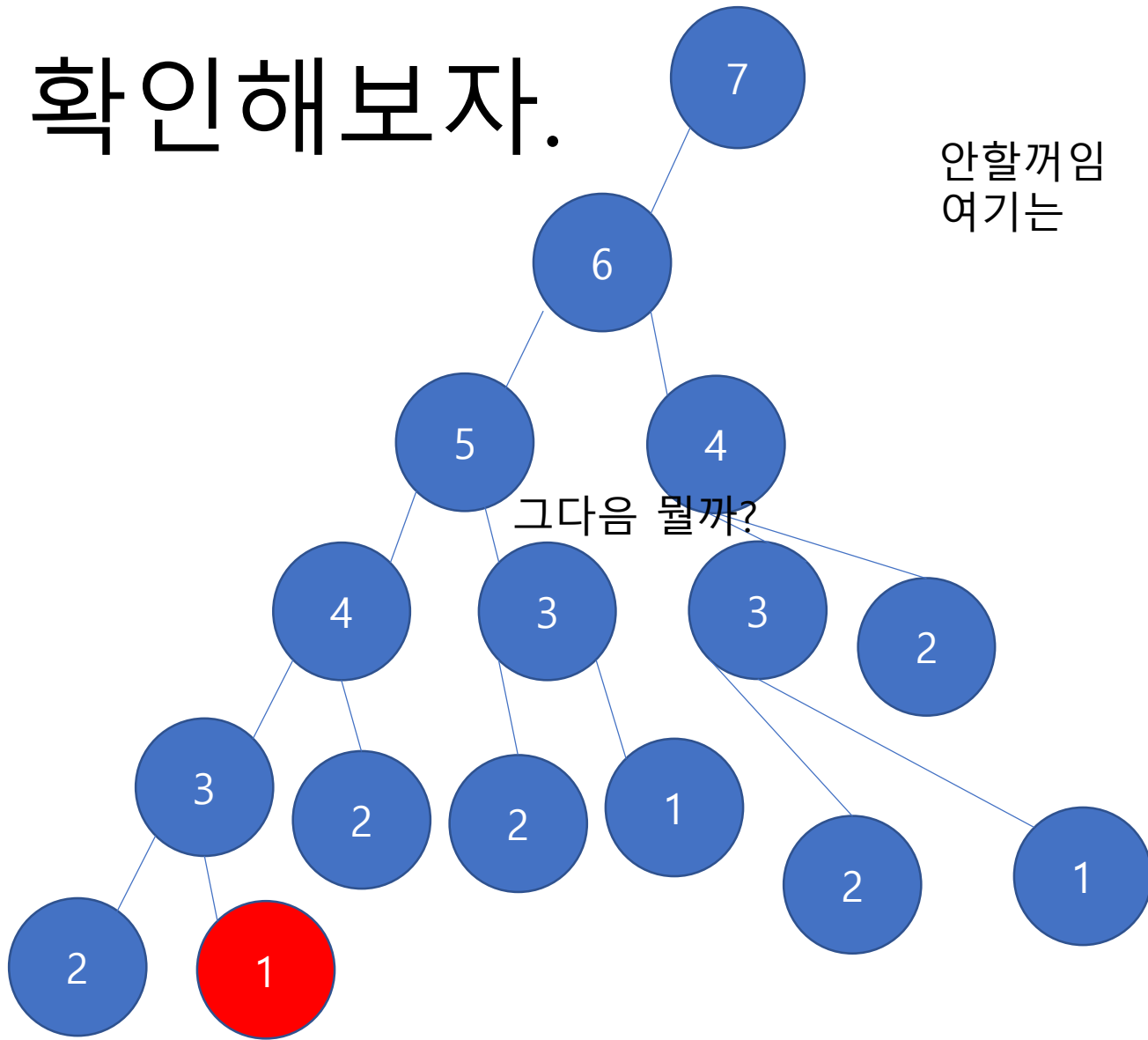
Fibo(7)을 호출했을 때, Fibo는 얼마나 호출이 될까?

# 확인해보자.

- 틀린 것이 당연하다. 하지만 지금 바로 고쳐야 내년에 고생 안 한다.
- 재귀는 Stack과 연관이 있다. 그런데 Stack은 트리 구조와 연관 있다.
- 이 마인드로 Fibo(7)을 다시 보자.

# 확인해보자.

안할꺼임  
여기는



## 그다음 뭘까?

```
int Fibo(int n) {
    printf("현재 호출된 함수 %d\n", n);
    if (n == 1) return 0;
    else if (n == 2) return 1;
    else return fibo(n - 1) + fibo(n - 2);
}
```



# 부탁

- 트리구조로 fibo함수를 같이 뜯어 보았다.
- 이제 집에 가서 오늘 했던 트리 구조로 '직접' 그려가면서 다시 복습하고 더 나아가 stack 구조로 새롭게 이해해 보자.
- 언제 함수가 return되는지 직접 디버깅을 해서 납득 될 때까지 복습해보자.
- 위 과정이 제대로 되지 않는다면 설 끝나고 '저.. 알고리즘이랑 안 맞는 거 같아요..' 라는 이야기가 나온다...

# 재귀 작성할 때 사고

- 반드시 종료조건을 명시해야 한다.
- 호출 스택에서 return 되면 처음으로 가는 것이 아니다. 그 함수가 호출된 곳으로 간다.
  - 그래서  $\text{fibo}(n-2)$ 이 계속 호출되다가  $\text{fibo}(n-1)$ 이 호출되는 것이다.
  - 이해가 가지 않는다면 이해하려고 노력하자.
- 터지는 것도 생각하자.
  - 터지지 않는 재귀인 꼬리 재귀라는 친구도 있다.

# 마무리

- 처음 보면 진짜 이해하기 힘들지만, 완전히 이해하면 ㅎㅎ..
- '재귀는 알아서 해결해주는 친구' 라는 생각이 들 정도로 재귀연습을 하자.
- 모르는 것이 있으면 물어보자!

# 스택 - STL

- <https://www.acmicpc.net/problem/10828> – 스택
- 직접 코딩하는 것을 같이 보자!

# 큐 - STL

- <https://www.acmicpc.net/problem/10845> – 큐
- 직접 STL을 이용해서 작성 해 보자.