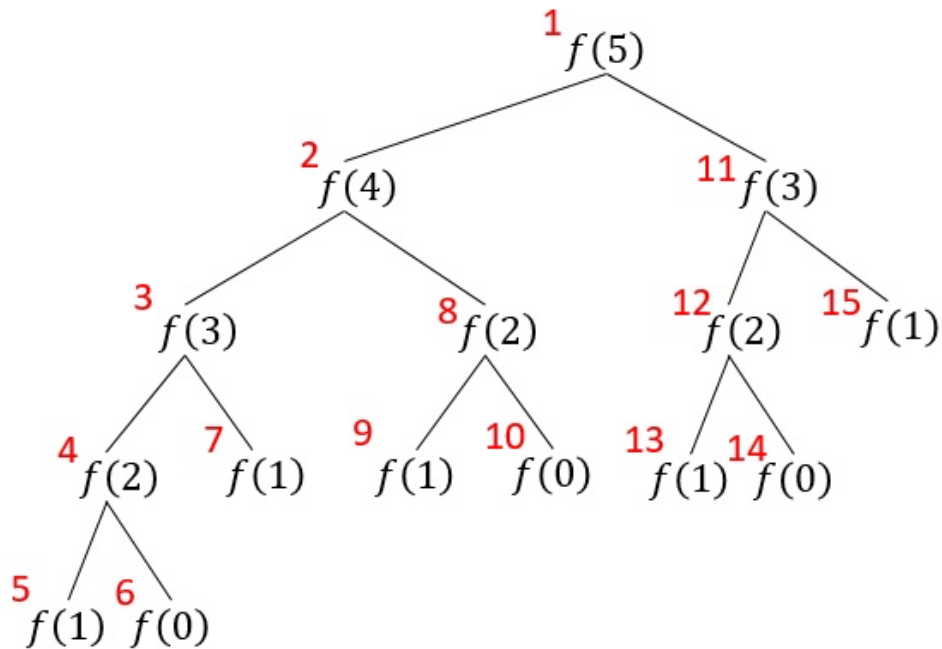


## 동적 계획법(Dynamic Programming)



동적 계획법 (Dynamic Programming)은 큰 문제를 작은 하위 문제로 나뉘서 푸는 알고리즘 설계 기법입니다. 이 때 중복되는 하위 문제를 한 번만 푸는 메모이제이션 (Memoization)이라는 기법을 사용하여 연산 속도를 향상시킵니다. 대표적인 예로는 피보나치 수열이 있습니다.

### DP ( with Javascript )

```
function fibonacci(n) {  
  const memo = [0, 1]; // 기초 케이스 값 초기화  
  
  for (let i = 2; i <= n; i++) {  
    memo[i] = memo[i - 1] + memo[i - 2]; // 이전 두 값의 합을 현재 값으로 저장  
  }  
  
  return memo[n]; // n번째 피보나치 수열 값 반환  
}  
  
console.log(fibonacci(10)); // 55
```

위 코드에서는 memo 배열을 선언하여 피보나치 수열의 기초 케이스 값을 저장합니다. 이후 반복문을 통해 이전 두 값의 합을 현재 값으로 계산하여 memo 배열에 저장합니다. 이렇게 되면 중복되는 계산을 한 번만 수행하므로 연산 속도가 향상됩니다. 마지막으로 n번째 피보나치 수열 값을 반환합니다.