분할 정복 알고리즘: 해결할 수 없는 문제를 작은 문제로 분할하여 문제를 해결하는 방법/알고리즘 -> 재귀함수(recursive function)를 통해 구현

#크고 방대한 문제를 해결할 때 자주 사용

#둘 이상의 부분 문제로 나눌 수 있는 자연스러운 방법과 부분 문제의 답을 조합해 원래 문제의 답을 계산하는 효율적인 방법이 존재해야 함

#빠르고 편리하게 작업을 처리 가능하지만/ 재귀 함수 사용 시 함수 호출 오버헤드 때문에 오히려 실행 속도가 늦어질 수 있음, 스택 오버플로우가 발생하거나 과도하게 메모리를 사용하게 됨 -> stack이나 queue 등을 통하여 분할 정복을 구현 가능

구현:

1. Divide – 문제 분할이 가능할 경우 2개 이상의 문제로 나눔
2. Conquer – 나누어진 문제가 분할이 가능할 경우 divide를 수행, 더 이상 분할하지 않고 해결이 가능한 작은 문제에 도달했을 경우 문제 풀기
3. Combine – Conquer한 문제들을 통합

function F(x):

if F(x)의 문제가 간단 then:

return F(x)을 직접 계산한 값

else:

x를 y1, y2로 분할

F(y1)과 F(y2)를 호출

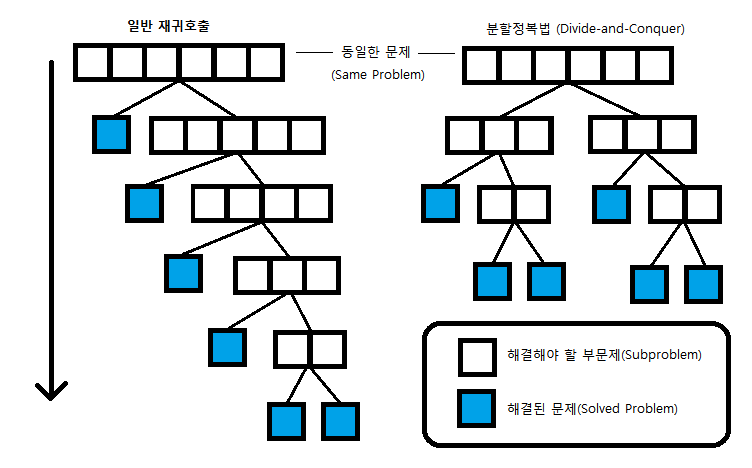
return F(y1), F(y2)로부터 F(x)를 구한 값

위와 같은 알고리즘으로 구현: F(x)가 간단해질 때까지 함수를 쪼갬

사진-



\*일반 재귀 알고리즘과의 차이:



Merge sort, Exponentiation, Fibonacci sequence, 이분검색, 최대값 찾기, 임계값의 결정, 쉬트라센 행렬곱셈 등을 분할 정복으로 구현 가능