

# Dynamic programming

동적 프로그래밍 알고리즘

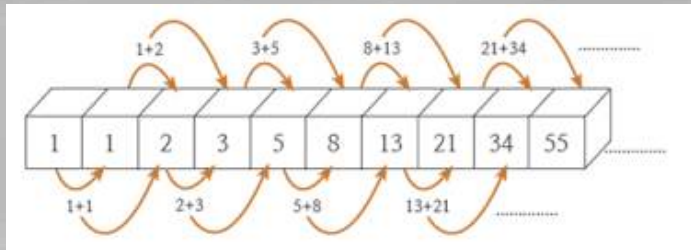
# 다이나믹 프로그래밍이란

- 큰 문제를 작은 문제로 나누어 푸는 법
- 메모리를 적절히 사용하여 수행 시간 효율성을 비약적으로 향상시킨다.
- 이미 계산된 결과는 별도의 메모리 영역에 저장하여 다시 계산하지 않도록 한다.
- 탑다운과 바텀업으로 크게 2가지로 나눌 수 있다.
- 동적프로그래밍?

# 다이나믹 프로그래밍의 조건

1. 큰 문제를 작은 문제로 나눌 수 있으며 작은 문제의 답을 모아  
서 큰 문제를 해결할 수 있을 때
2. 동일한 작은 문제를 반복적으로 해결해야 할 때

## 예시-피보나치 수열



$$F_n = F_{n-1} + F_{n-2}$$

$$F_1 = F_2 = 1$$

# 피보나치 수열 소스코드(C언어)

main.c x

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int fibo(int x)
5  {
6      if(x==1 || x==2)
7          return 1;
8      return fibo(x-1)+fibo(x-2);
9  }
10
11 int main()
12 {
13     int n;
14     printf("몇번째 자리를 출력하시겠습니까?");
15     scanf("%d", &n);
16     printf("%d", fibo(n));
17 }
18
```

# 결과값

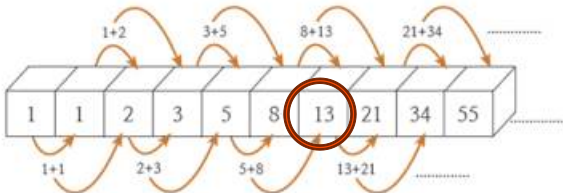
"C:\Users\boyman\OneDrive\바탕 화면\프로그래밍\c언어\fibonacci\bin\Debug\fibonacci.exe"

몇번째 자리를 출력하시겠습니까?

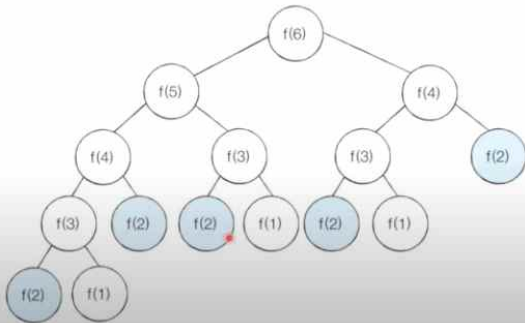
13

Process returned 0 (0x0) execution time : 6.323 s

Press any key to continue.



# 시간 복잡도 분석



$f(2)$ 가 여러 번 호출되어 중복이 되는 문제를 갖는다.

# 시간 복잡도 분석

- 따라서 시간 복잡도는  $O(2^n)$ 이 된다. 만약 30번째의 피보나치 수열값을 얻으려면  $2^{30}$ 인 대략 10억번의 연산을 수행해야 한다.



# 메모이제이션(Memoization)

- 컴퓨터 프로그램이 동일한 계산을 반복해야 할 때, 이전에 계산한 값을 메모리에 저장하여 동일한 계산의 반복 수행을 제거하는 기술이다.
- 값을 기록해 놓는다는 점에서 캐싱(Caching)이라고도 불린다.

# 탐다운과 바텀업

- Top-down은 가장 큰 문제를 방문 후 작은 문제를 호출 하여 답을 찾는 방식이다.
- Bottom-up은 가장 작은 문제들 부터 답을 구해가며 전체 문제의 답을 찾는 방식이다.
- 탐다운방식은 하향식이라고도 하며, 메모이제이션을 쓰고 바텀업은 상향식이라고 불린다.
- 다이나믹 프로그래밍의 전형적인 형태는 바텀업이다.

# 피보나치수열 탑다운방식(C언어)

main.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  long long int dp[100];
4
5  long long int fibo(int x)
6  {
7      if(x==1 || x==2)
8          return 1;
9      if(dp[x] != 0)
10         return dp[x];
11     dp[x] = fibo(x-1) + fibo(x-2);
12     return dp[x];
13 }
14 int main()
15 {
16     int n;
17     printf("몇번째자리의 값을 출력하겠습니까?");
18     scanf("%d", &n);
19     printf("%lld", fibo(n));
20 }
21
```

# 출력결과

"C:\Users\boyima\OneDrive\바탕 화면\프로그래밍\c언어\fibonacci\bin\Debug\fibonacci.exe"

몇번째자리의 값을 출력하겠습니까?50

12586269025

Process returned 0 (0x0) execution time : 0.706 s

Press any key to continue.

## 피보나치 수열 생성기

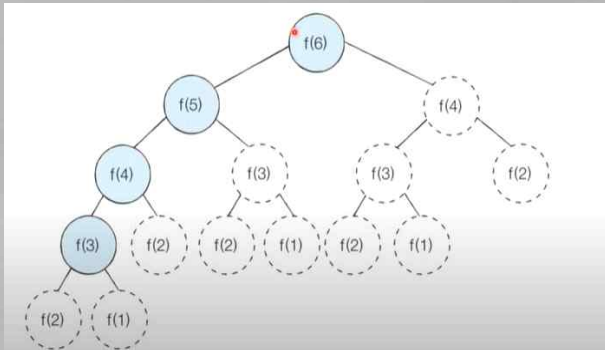
음수가 아닌 정수를 입력하세요.:

한도: 10000

피보나치 번호 50:

12586269025

# 메모이제이션 분석



시간복잡도는  $O(n)$ 이 된다.

# 피보나치수열 바텀업방식(C언어)

main.c x

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  long long int dp[100];
4  int main()
5  {
6      dp[1]=1;
7      dp[2]=1;
8      int n;
9      printf("몇번째 자리를 출력할까요?");
10     scanf("%d", &n);
11     for(int i=3; i<=n; i++)
12     {
13         dp[i]=dp[i-1]+dp[i-2];
14     }
15     printf("%lld", dp[n]);
16 }
17
```

# 출력결과

 "C:\Users\boyoma\OneDrive\바탕 화면\프로그래밍\c언어\fibonacci\bin\Debug\fibonacci.exe"

몇번째 자리를 출력할까요?50

12586269025

Process returned 0 (0x0) execution time : 1.839 s

Press any key to continue.



QnA