

4 주차

DP


DP(Dynamic Programming)?

피보나치수₂

- <https://www.acmicpc.net/problem/2748>

$$F(n) := \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

```
1  #include <stdio>
2  using namespace std;
3
4  int fibonacci(int n){
5      if(n == 0) return 0;
6      if(n == 1) return 1;
7      return fibonacci(n-2) + fibonacci(n-1);
8  }
9
10 int main(){
11     int N;
12     scanf("%d", &N);
13     printf("%d\n", fibonacci(N));
14 }
```

Colored by Color Scripter 

Top-Down

```
1  #include <stdio>
2  #include <vector>
3  using namespace std;
4
5  vector<int> dp;
6
7  int fibonacci(int n){
8      if(n == 0) return 0;
9      if(n == 1) return 1;
10     // 이미 값을 계산한 적이 있다면 그 값을 바로 리턴
11     if(dp[n] != -1) return dp[n];
12     // 아니라면 계산해서 dp 리스트에 넣어 보존
13     dp[n] = fibonacci(n-2) + fibonacci(n-1);
14     return dp[n];
15 }
16
17 int main(){
18     int N;
19     scanf("%d", &N);
20     dp.resize(N+1, -1); // 초기값 -1은 fibonacci 결과로 절대 나올 수 없는 값
21     printf("%d\n", fibonacci(N));
22 }
```

Colored by Color Scripter

Bottom-Up

```
1  #include <cstdio>
2  #include <vector>
3  using namespace std;
4
5  int main(){
6      int N;
7      scanf("%d", &N);
8      vector<int> dp(N+1, 0);
9      dp[1] = 1;
10     for(int i=2; i<=N; i++)
11         dp[i] = dp[i-1] + dp[i-2];
12     printf("%d\n", dp[N]);
13 }
```

DP실습(1) – 1로 만들기

- <https://www.acmicpc.net/problem/1463>

DP실습(2) – 스티커

- <https://www.acmicpc.net/problem/9465>

DP실습(3) – 2xn 타일링

- <https://www.acmicpc.net/problem/11726>

DP실습(4) – 동전 2

- <https://www.acmicpc.net/problem/2294>

DP실습(5) – 제곱수의 합

- <https://www.acmicpc.net/problem/1699>

DP실습(6) – 쉬운 계단수

- <https://www.acmicpc.net/problem/10844>

DP실습(7) – 이항 계수 2

- <https://www.acmicpc.net/problem/11051>

DP실습(8) – 가장 큰 증가하는 부분수열

- <https://www.acmicpc.net/problem/11055>