

# 2023 BP

# 메모이제이션과 동적 계획법

# 메모이제이션 memoization

- 메모이제이션은 동일한 계산을 반복해야 하는 경우  
계산했던 결과를 저장해 뒀다가 필요할 때 꺼내 쓰는 기법입니다.
- 메모리 공간을 소모해서 실행 시간을 단축합니다.
- 반복 계산이 잦을 수록 유리한 프로그램을 짤 수 있습니다.

# 메모이제이션 사용 유무 비교

- 메모이제이션X

```
#include<stdio.h>

long long int f(int n){
    if(n<=2) return 1;
    return f(n-1)+f(n-2);
}

int main(){
    int n; scanf("%d", %n);
    printf("%lld", f(n));
}
```

- 메모이제이션O<sub>top-down</sub>

```
#include<stdio.h>

long long int d[100]={0,1,1};
long long int f(int n){
    if(d[n]) return d[n];
    return f(n-1)+f(n-2);
}

int main(){
    int n; scanf("%d", %n);
    printf("%lld", f(n));
}
```

- 메모이제이션O<sub>bottom-up</sub>

```
#include<stdio.h>

long long int d[100]={0,1,1};

int main(){
    int n; scanf("%d", %n);
    for(int i = 2; i <= n; i++){
        d[i] = d[i-1]+d[i-2];
    }
    printf("%lld", d[n]);
}
```

# 동적 계획법 Dynamic Programming

- 어떤 값을 구하기 위해서 그 이전에 구한 값들을 재사용하는 기법입니다.
- 한 문제를 그보다 작은 문제의 연장선으로 생각하고 과거의 해를 재사용하는 해결 방식을 총칭합니다.
- 일반적으로, 점화식을 통해 문제의 답을 구할 수 있는 경우에 활용됩니다.

문제	난이도
<p><b>A</b>    알고리즘 수업 – 피보나치 수 1</p> <p><b>B</b>    타일 장식물</p> <p><b>C</b>    피보나치 비스무리한 수열</p> <p><b>D</b>    1, 2, 3 더하기</p>	<p><b>Bronze I</b></p> <p>Silver V</p> <p>Silver IV</p> <p>Silver III</p>

BOJ 24416

# 알고리즘 수업 – 피보나치 수 1

#dp

난이도 – Bronze I

## BOJ 24416 알고리즘 수업 – 피보나치 수 1

- 메모이제이션을 잘 활용하면 얼마나 효율적으로 코드를 짤 수 있는 지 확인할 수 있는 문제입니다.
- 첫 번째 코드는 재귀적으로 피보나치 수를 구합니다.
- 첫 코드의 #코드1 부분은 실행될 때마다 1을 반환합니다.  
이때 반환된 1은 누적되어 최종적으로 N번째 피보나치 수가 됩니다.
- 즉 #코드1은 N번째 피보나치 수만큼 실행됩니다.

## BOJ 24416 알고리즘 수업 – 피보나치 수 1

- 두 번째 코드는 메모이제이션을 활용합니다.
- $f[k]$ 에  $k$ 번째 피보나치 수를 저장하는 것으로  
매번  $k$ 번째 피보나치 수를 구하는 번거로운 과정을 피할 수 있습니다.
- #코드2가 실행될 때마다  $f[3]$ 부터 순서대로 피보나치 수가 구해집니다.
- 최종적으로  $f[N]$ 을 구해야 하므로 #코드2는  $N-2$ 번 실행됩니다.



# BOJ 13301

# 타일 장식물

#dp

난이도 – Silver V

## BOJ 13301 타일 장식물

- 와 피보나치 수열이다!
- $f[1] = f[2] = 1$ ,  $f[k] = f[k-1] + f[k-2]$  ( $k \geq 3$ ) 입니다.
- 메모이제이션을 이용하면 빠른 시간 안에 구할 수 있습니다.
- 직사각형의 둘레는  $2 \times (f[N+1] + f[N]) = 2 \times f[N+2]$  입니다.

BOJ 14495

# 피보나치 비스무리한 수열

#dp

난이도 – Silver IV

## BOJ 14495 피보나치 비스무리한 수열

- 문제에서 점화식과 초항이 모두 주어졌습니다.
- 동적 계획법을 이용해 문제를 해결합니다!
- 단, int 사용 시 범위를 넘어갈 수 있습니다.  
long long int를 사용해서 해당 문제를 해결할 수 있습니다.

BOJ 9095

1, 2, 3 더하기

#dp

난이도 – Silver III

## BOJ 9095 1, 2, 3 더하기

- 동적 계획법을 쓰기 위해서는 점화식이 필요합니다.  
점화식을 찾는 방법을 연습해 봅시다.
- 초항과 일반항을 구해야 합니다. 어디부터가 초항이고 일반항인 걸까요??

## BOJ 9095 1, 2, 3 더하기

- $N$ 을 만드는 방법을 생각해 봅시다.
- 할 수 있는 행동이 1, 2, 3 더하기 뿐이므로, 가능한 경우는  $(N-1)+1$ ,  $(N-2)+2$ ,  $(N-3)+3$  꼴 뿐입니다.
- 즉  $N$ 을 만드는 경우의 수는  $N-1$ ,  $N-2$ ,  $N-3$ 을 만드는 경우의 수의 합이 됩니다.

## BOJ 9095 1, 2, 3 더하기

- 근데 이러면  $N$ 이 3 이하일 때는  $N-3$ 을 만드는 방법을 알 수가 없네요.
- 이럴 땐  $N$ 이 1, 2, 3일 때를 직접 계산해서 미리 저장해 두면 됩니다.
- $N=1$ 인 경우: (1) 의 1가지
- $N=2$ 인 경우: (2), (1, 1)의 2가지
- $N=3$ 인 경우: (3), (2, 1), (1, 2), (1, 1, 1)의 4가지



## BOJ 9095 1, 2, 3 더하기

- 이를 정리하면 다음과 같습니다.
- $d[N]$  = 1, 2, 3을 더해서 N을 만드는 방법의 수 (순서를 고려함)
- $d[1] = 1, d[2] = 2, d[3] = 4$
- $d[k] = d[k-1] + d[k-2] + d[k-3] \ (k \geq 4)$