

# 동적계획법

## Dynamic Programming

최아현

# 동적계획법

: 이미 했던 연산이 반복되는 결점을 보완하기 위한 방법

처음 진행되는 연산을 기록해두고, 이미 진행했던 연산이라면 다시 연산하기 보단 기록되어 있는 값을 가져오는 기법

※ 메모이제이션: 메모이제이션은 동일한 계산을 반복해야 할 때, 이전에 계산한 값을 저장해서 동일한 계산의 반복 수행을 제거하는 기술

시간과 메모리의 trade-off

# 분할정복 기법

: 큰 문제를 한번에 해결하기 힘들 때 작은 여러 개의 문제로 나누어서 푸는 기법

작은 문제를 풀다보면 같은 문제들을 반복해서 푸는 경우가 생김

-> 그 문제들을 매번 재계산하지 않고 값을 저장해두었다가 재사용하는 기법이 동적 프로그래밍

# 동적 계획법의 적용 조건

최적 부분구조 (optimal substructure)

- 큰 문제의 최적 솔루션에 작은 문제의 최적 솔루션이 포함됨

재귀호출 시 중복 (overlapping recursive calls)

- 재귀적으로 구현했을 때 중복 호출로 심각한 비효율이 발생

# Ex) 피보나치 수열

fibonacci(10) 를 호출하면 점점 작은수를 호출하면서  
가장작은수 2 까지 도달하는 방식

```
int fiboData[100] = { 0, };  
int fibo(int n) {  
    if (n <= 2)  
        return 1;  
    if (fiboData[n] == 0) // 배열에 값이 없다면  
        fiboData[n] = fibo(n - 1) + fibo(n - 2); // 값을 넣어줌  
    return fiboData[n]; // 있다면 return  
}
```

# 11053번 - 가장 긴 증가하는 부분 수열 (실버3)

## 문제

수열 A가 주어졌을 때, 가장 긴 증가하는 부분 수열을 구하는 프로그램을 작성하시오.

예를 들어, 수열 A = {10, 20, 10, 30, 20, 50} 인 경우에 가장 긴 증가하는 부분 수열은 A = {10, 20, 10, 30, 20, 50} 이고, 길이는 4이다.

## 입력

첫째 줄에 수열 A의 크기 N ( $1 \leq N \leq 1,000$ )이 주어진다.

둘째 줄에는 수열 A를 이루고 있는  $A_i$ 가 주어진다. ( $1 \leq A_i \leq 1,000$ )

## 출력

첫째 줄에 수열 A의 가장 긴 증가하는 부분 수열의 길이를 출력한다.

### 예제 입력 1 복사

```
6
10 20 10 30 20 50
```

### 예제 출력 1 복사

```
4
```

# 코드

```
#include <cstdio>
#include <iostream>
#include <algorithm>

using namespace std;

int num[1001] = { 0, };
int dp[1001] = { 0, };

int main() {
    int N, cnt = 0;
    cin >> N;

    for (int i = 1; i <= N; i++) {
        cin >> num[i];
    }

    for (int i = 1; i <= N; i++) {
        for (int j = 0; j < i; j++) {
            if (num[j] < num[i]) {
                dp[i] = max(dp[i], dp[j] + 1);
            }
            if (cnt < dp[i]) {
                cnt = dp[i];
            }
        }
    }

    cout << cnt;

    return 0;
}
```

$i = 1, \text{num}[i] = 10$

num	0	10	20	10	30	20	50
-----	---	----	----	----	----	----	----

dp	0	1	0	0	0	0	0
	[0]	[1]	[2]	[3]	[4]	[5]	[6]

cnt = 1



$i = 2, \text{num}[i] = 20$

num	0	10	20	10	30	20	50
-----	---	----	----	----	----	----	----

dp	0	1	2	0	0	0	0
	[0]	[1]	[2]	[3]	[4]	[5]	[6]

cnt = 2

$i = 3, \text{num}[i] = 10$

num	0	10	20	10	30	20	50
-----	---	----	----	----	----	----	----

dp	0	1	2	1	0	0	0
	[0]	[1]	[2]	[3]	[4]	[5]	[6]

cnt = 2

$i = 4, \text{num}[i] = 30$

num	0	10	20	10	30	20	50
-----	---	----	----	----	----	----	----

dp	0	1	2	1	3	0	0
	[0]	[1]	[2]	[3]	[4]	[5]	[6]

cnt = 3

$i = 5, \text{num}[i] = 20$

num	0	10	20	10	30	20	50
-----	---	----	----	----	----	----	----

dp	0	1	2	1	3	2	0
	[0]	[1]	[2]	[3]	[4]	[5]	[6]

cnt = 3

$i = 6, \text{num}[i] = 50$

num	0	10	20	10	30	20	50
-----	---	----	----	----	----	----	----

dp	0	1	2	1	3	2	4
	[0]	[1]	[2]	[3]	[4]	[5]	[6]

cnt = 4

# 문제

[백준] 11053번 - 가장 긴 증가하는 부분 수열 (실버3)

[백준] 2579번 - 계단오르기(실버3)

[백준] 12865번 - 평범한 배낭(골드5)

[백준] 9251번 - LCS(골드5)

[백준] 2169번 - 로봇 조종하기(골드1)