DP - Dynamic Programming

서론

> 💡 Dynamic과 아무 연관이 없다!

보통 Dynamic(동적) 이라고 하면 런타임중 실시간으로 변한다거나 아주 멋진 프로그래밍이 있을거라 생각하게 됩니다. 그런데 실제로 이런 내용들이 있는 개념은 아니고 이 알고리즘의 창시자인 `벨만` 이 **멋있어서** Dynamic 이라고 이름을 붙이게 되었습니다.

서울대학교 컴퓨터공학부 이광근 교수님은 번역에서 동적 계획법이 아니라 `기억하며 풀기`로 번역을 했는데 이쪽의 어감이 더 DP 의 특성과 맞다고 생각됩니다.

피보나치 수의 예시

만일 피보나치 수를 구한다고 하면 아래와 같은 식이 주로 나옵니다.

```
int fib(int a){
  if(a <= 1)
    return 1;
  return fib(a-1) + fib(a-2);
}
```

위 식의 문제점이라고 하면 a 가 커질때 상상 이상으로 시간이 오래 걸리게 되어 알고리즘 문제를 풀때는 적합하지 않은 방식이 됩니다.

```
int fibNum[10000];
fibNum[0] = 1;
fibNum[1] = 1;

void fib(int a){
  for(int i = 2; i <= a; i++){
    fibNum[i] = fibNum[i-1] + fibNum[i-2];
  }
}

int main(){
  int input;
```

```
  cin >> input;

  fib(input);

  cout << fibNum[input];
}
```

대신 이런 식으로 배열에 저장을 하면서 풀이하게되면 시간을 O(N)으로 줄일 수 있게 됩니다.

DP의 특징

분할 가능한 문제

- 피보나치 수의 예시처럼 큰 문제가 있을때 이를 작은 문제들로 분할이 가능할때 사용하게 됩니다.
- 작은 문제로 분할을 해도 문제의 전체적인 본질을 흐리지 않아야 합니다.

중복되는 부분 문제들

- 단순 분할정복과는 다르게 **중복되는 부분들**이 등장하게 됩니다.
- 이를 배열 등에 저장을 해서 문제 풀이를 하는게 주 목적입니다.

결과는 확실하게 보장

- 그리디와는 다르게 완전탐색의 일종으로 확실한 결과를 보장합니다.

추천 문제

11726번: 2×n 타일링

2×n 크기의 직사각형을 1×2, 2×1 타일로 채우는 방법의 수를 구하는 프로그램을 작성하시오.

BAEKJOON ONLINE JUDGE

https://www.acmicpc.net/problem/11726

12865번: 평범한 배낭

BAEKJOON ONLINE JUDGE

https://www.acmicpc.net/problem/12865

2096번: 내려가기

문제 N줄에 0 이상 9 이하의 숫자가 세 개씩 적혀 있다. 내려가기 게임을 하고 있는데, 이 게임은 첫 줄에서 시작해서 마지막 줄에서 끝나게 되는 놀이이다. 먼저 처음에 적혀 있는 세 개의 숫자 중에서 하나를 골

https://www.acmicpc.net/problem/2096