

9주차: DP (Dynamic Programming)

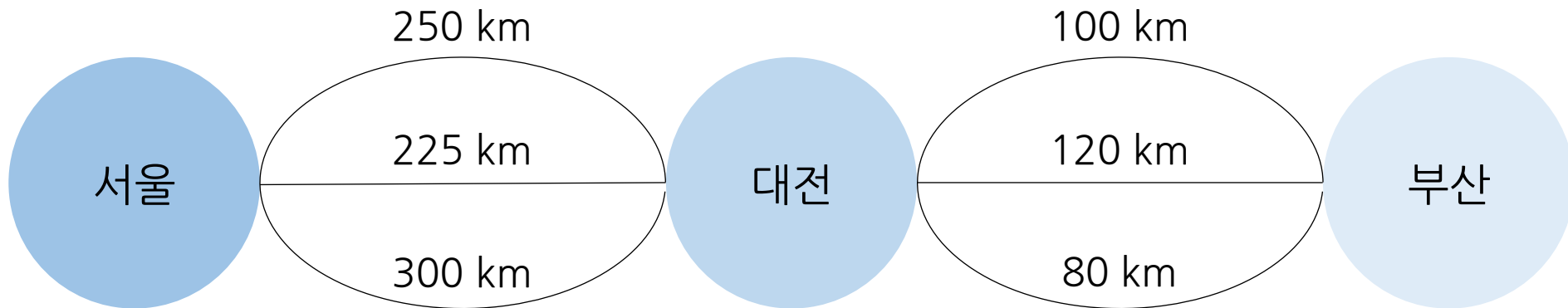
```
33 self.fingerprints = {}
34 self.logdups = True
35 self.debug = debug
36 self.logger = logging.getLogger(__name__)
37 if path:
38     self.file = open(os.path.join(path, 'requests.json'), 'w')
39     self.file.seek(0)
40     self.fingerprints.update({})
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.get('DEBUG', False)
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     fingerprint = fingerprint(request)
```

커리큘럼

1. 변수, 입출력
2. 조건문, 반복문
3. 리스트, 튜플, 세트, 딕셔너리
4. 함수
5. class
6. 알고리즘 입문, 그리디 알고리즘
7. 재귀함수
8. 탐색
9. DP(Dynamic Programming)

DP(Dynamic Programming)

- ✓ Richard E. Bellman 이 제안
- ✓ “최적 부분 구조”를 가진 문제를 해결하는 데 사용



접근 방법

1. 분할 정복의 아이디어: 문제를 분할해서 생각
 2. 작은 문제들을 순차적으로 풀어나간다.
 3. 풀어가면서 결과를 하나씩 다 저장해 나가고 그 다음 단계는 앞선 결과를 활용하여 점화식을 세우는 등으로 풀어나간다(Memorization)
-
- ✓ 단순한 재귀함수 알고리즘에 비해 이전 값을 활용하기에 빠르다.
 - ✓ 그리디 알고리즘보다는 느리고, 저장 공간을 많이 차지한다. 하지만 최적성에 있어서는 그리디 알고리즘보다는 좋은 알고리즘이다.

피보나치 수열

피보나치 수열의 항을 구하는 문제. n 번째 피보나치 수열의 항을 구하시오.

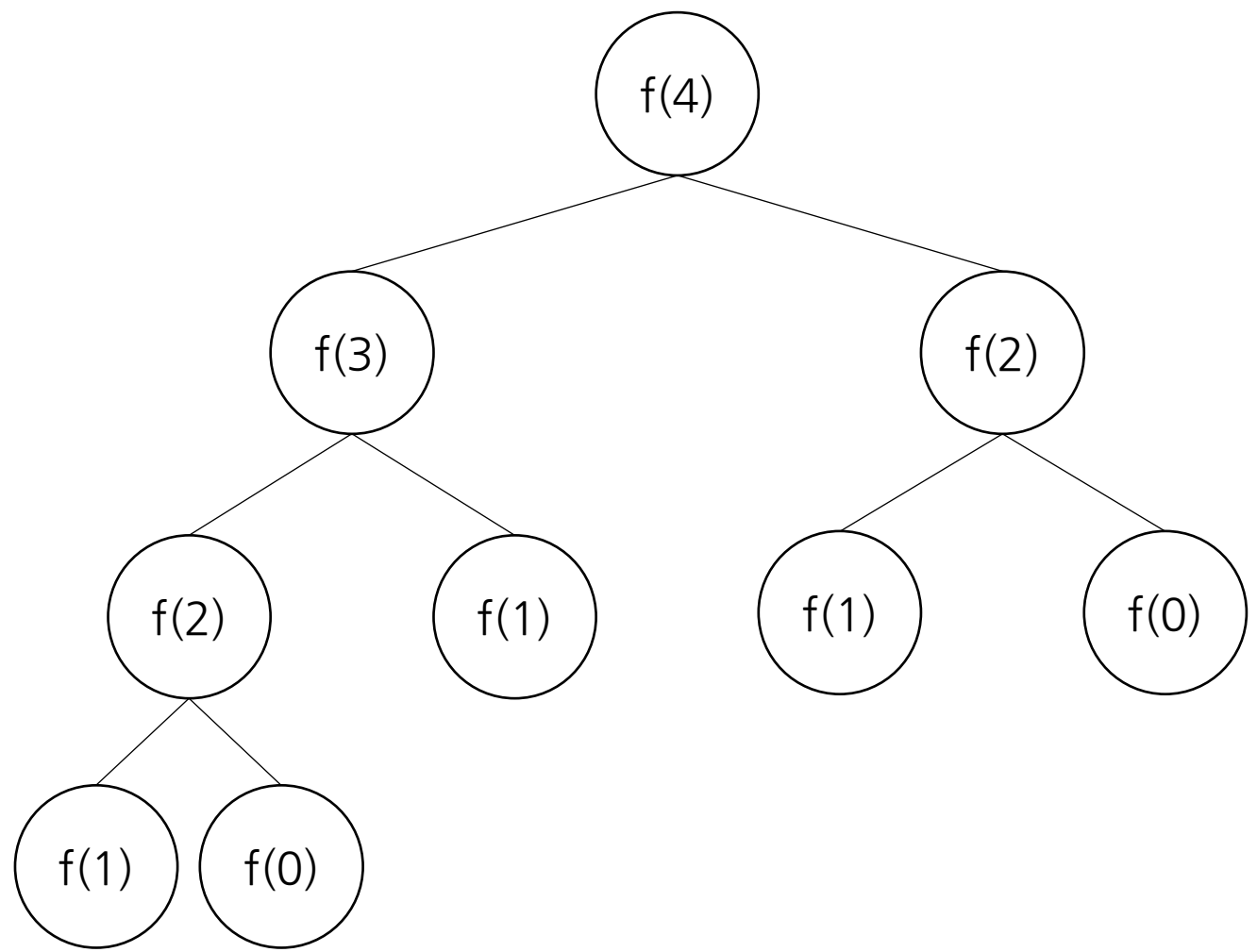
$$a_n = a_{n-1} + a_{n-2} \ (n \geq 2), a_0 = 1, a_1 = 1$$

```
n=int(input())
left=1
right=1
result=0
for i in range(n+1):
    if i==0 or i==1:
        result=1
        continue
    result=left+right
    right=left
    left=result
print(result)
```

```
def fibonacci(n):
    if n==0:
        return 1
    elif n==1:
        return 1
    else:
        return fibonacci(n-1)+fibonacci(n-2)

n=int(input())
a = fibonacci(n)
print(a)
```

피보나치 수열



Top-down



Bottom-up

피보나치 수열

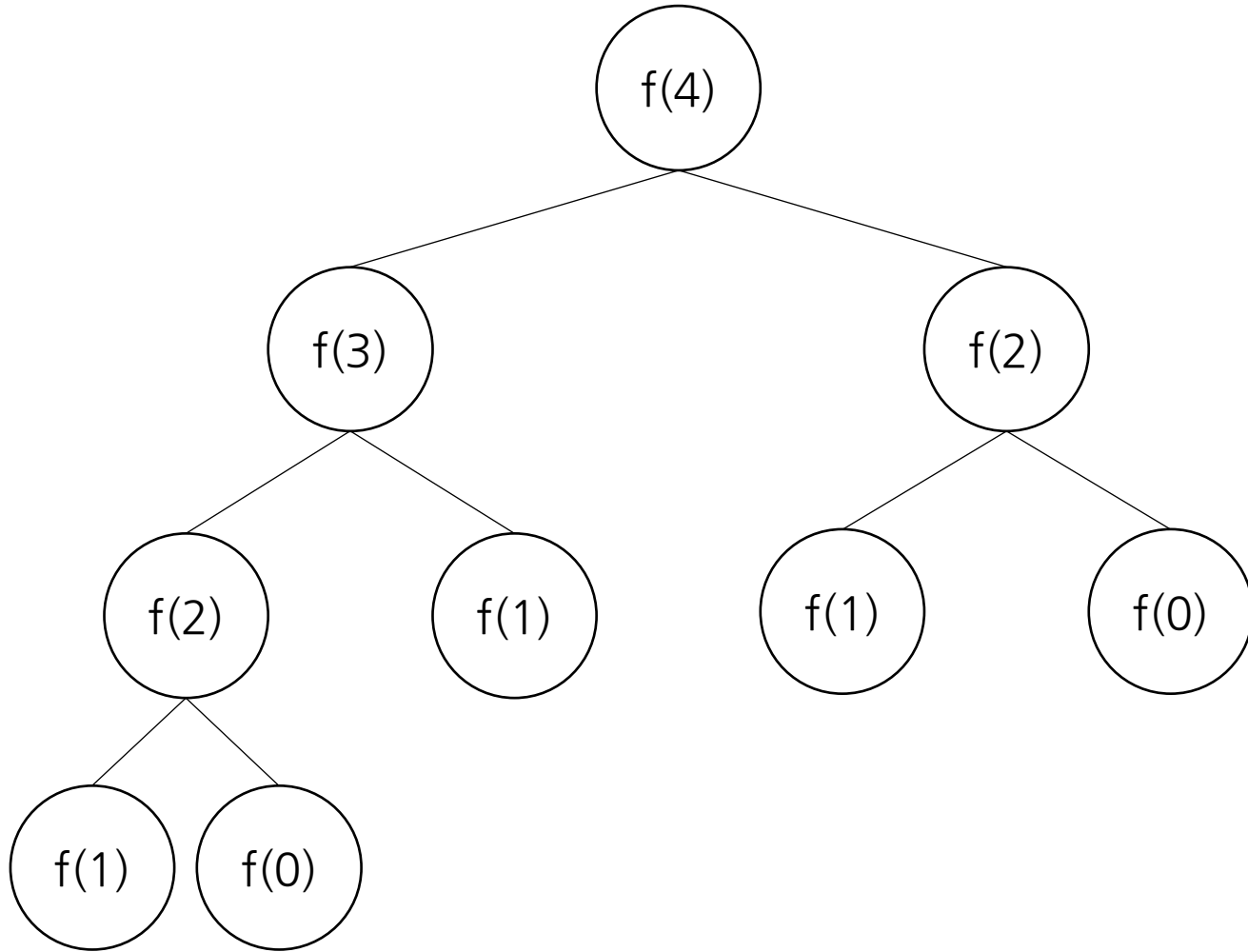
Top-down

```
memo = [0 for _ in range(100)]
def f(n):
    if n <= 1:
        return 1
    if memo[n] != 0:
        return memo[n]
    memo[n] = f(n-1)+f(n-2)
    return memo[n]
```

Bottom-up

```
memo = [0 for _ in range(100)]
memo[0], memo[1] = 1, 1
def f(n):
    if memo[n-1] == 0:
        for i in range(2, n+1):
            memo[i] = memo[i-1] + memo[i-2]
    return memo[n]
```

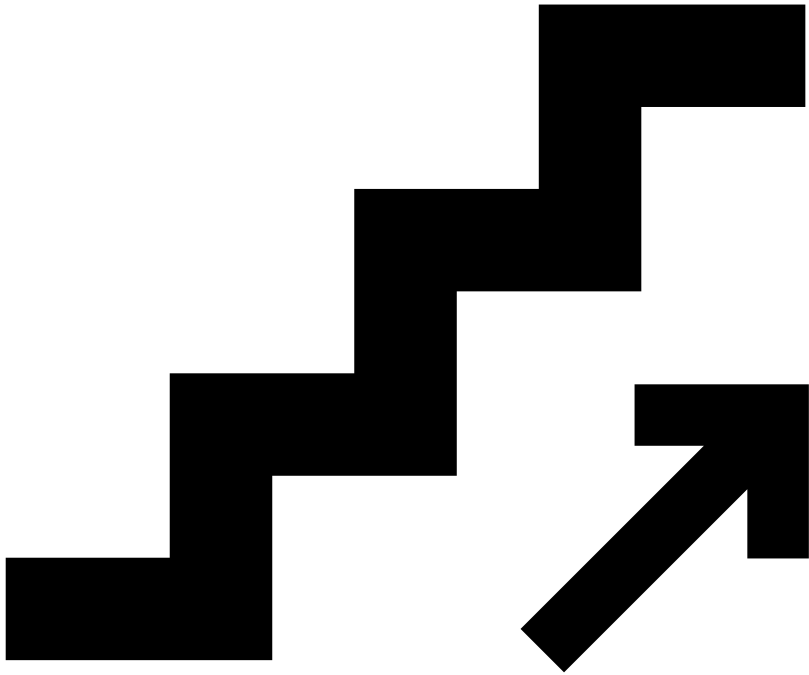
예제 1 (피보나치 수열 응용)



$f(n)$ 을 구하는 재귀함수 코드에서 $f(1)$ 과 $f(0)$ 이 각각 몇 번 수행되는지 DP를 활용해서 알아보기 바랍니다.

예를 들어, $f(4)$ 에서는 $f(1)$ 3번, $f(0)$ 2번이 수행됩니다.

예제2



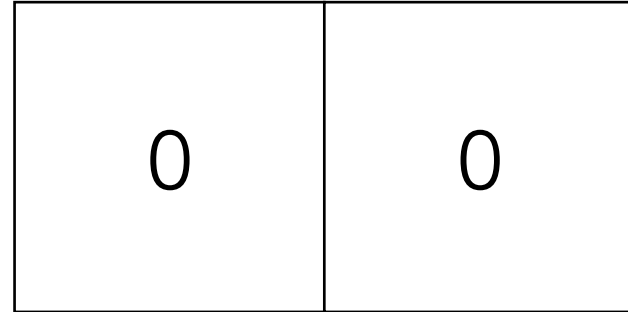
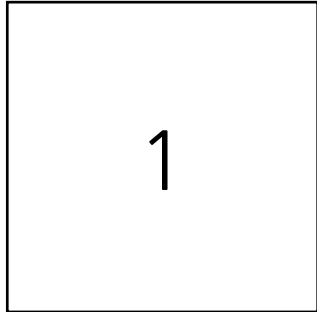
한번에 계단을 1개 또는 2개를 올라갈 수 있습니다.

이러한 방법으로 계단을 올라갈 때, n 개의 계단을 올라가는 방법을 알아보시기 바랍니다.

예제3

‘1’ 과 ‘0’을 두 개씩 붙인 ‘00’ 타일 총 두 종류의 타일이 있습니다. ‘00’ 타일은 분리할 수 없고 타일의 개수는 무한하다고 할 때, 이 타일들을 이어서 여러 수열을 만들고자 합니다. 이 때, 크기가 N인 수열을 만들 수 있는 가짓수를 알아보기 바랍니다.

예를 들어, N=1일 때, 1만 만들 수 있고, N=2일 때는 00, 11을 만들 수 있습니다. 또한 N=4일 때는 0011, 0000, 1001, 1100, 1111 총 5개의 수열을 만들 수 있습니다.



```
31 def __init__(self, path):
32     self.file = None
33     self.fingerprints = set()
34     self.logdups = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, 'requests.log'),
39                         'a')
40         self.file.seek(0)
41         self.fingerprints.update(e.request for e in self.requests)
```

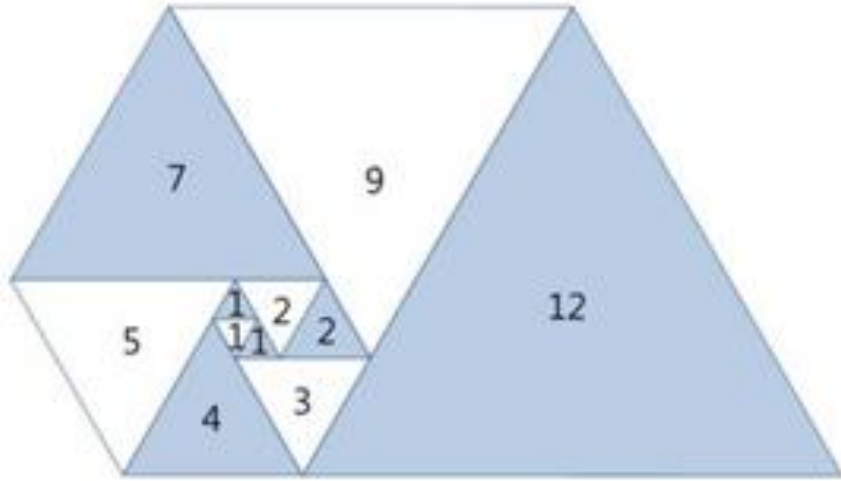
```
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getbool('DEBUG', False)
45     return cls(settings_dir(settings), debug)
```

```
46 def request_seen(self, request):
47     fp = self.request_fingerprint(request)
48     if fp in self.fingerprints:
49         return True
50     self.fingerprints.add(fp)
51     if self.file:
52         self.file.write(fp + os.linesep)
```

```
53 def request_fingerprint(self, request):
54     return request_fingerprint(request)
55
56
```

과제

9-1



왼쪽 그림과 같이 삼각형이 나선 모양으로 놓여져 있습니다. 첫 삼각형은 정삼각형으로 변의 길이는 1입니다. 그 다음에는 다음과 같은 과정으로 정삼각형을 계속 추가합니다. 나선에서 가장 긴 변의 길이를 k 라 했을 때, 그 변에 길이가 k 인 정삼각형을 추가합니다.

N 번째 정삼각형의 변의 길이를 구해보기 바랍니다.

10

입력 예시

9

출력 예시

9-2

번호가 매겨진 N개의 집을 페인트칠 하고자 합니다. 집은 빨강, 초록, 파랑 중 하나의 색으로 칠해야 하고, 각각의 집을 빨강, 초록, 파랑으로 칠하는 비용이 주어졌을 때, 아래 규칙을 만족하면서 모든 집을 칠하는 비용의 최솟값을 구해보기 바랍니다.

- ✓ 1번 집의 색은 2번 집의 색과 같지 않아야 한다.
- ✓ N번 집의 색은 N-1번 집의 색과 같지 않아야 한다.
- ✓ $i(2 \leq i \leq N-1)$ 번 집의 색은 $i-1$ 번, $i+1$ 번 집의 색과 같지 않아야 한다.

```
3
26 40 83
49 60 57
13 89 99
```

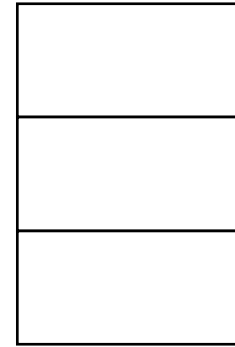
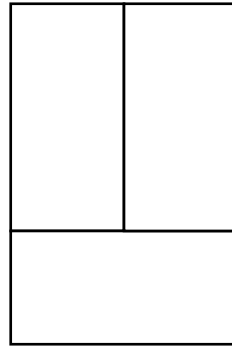
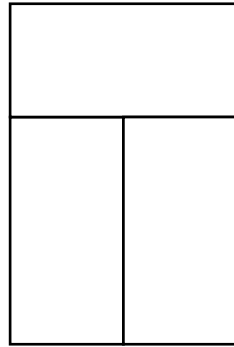
96

- ✓ 첫째 줄: N
- ✓ $k(k > 1)$ 번째 줄: $(k-1)$ 번 집을 각각 빨강, 초록, 파랑으로 칠하는 비용

출력 예시

9-3

N이 주어질 때, $3 \times N$ 크기의 벽을 2×1 , 1×2 크기의 타일로 채우는 경우의 수를 구해보기 바랍니다.



2

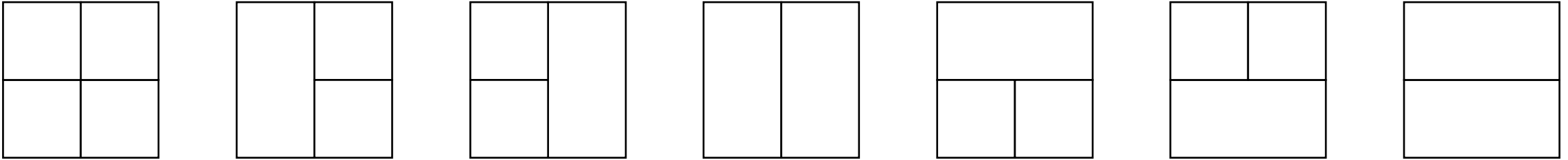
입력 예시

3

출력 예시

9-4

N이 주어질 때, $2 \times N$ 크기의 벽을 2×1 , 1×2 , 1×1 크기의 타일로 채우는 경우의 수를 구해보기 바랍니다.



2

입력 예시

7

출력 예시

9-5

수열 A가 주어졌을 때, 가장 긴 증가하는 부분 수열의 길이를 구하기 바랍니다.

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 4입니다.

10 20 10 30 20 50

입력 예시

4

출력 예시

9-6

어떤 상인이 배낭으로 채긴 물건을 팔려고 합니다. 자신이 가지고 있는 총 N 개의 물건에 대해서 각 물건은 무게 W 와 가치 V 를 가지는데, 배낭에는 최대 K 만큼의 무게만을 넣을 수 있습니다. 가치가 최대가 되고자 할 때, 가치의 최대값을 구해보기 바랍니다.

```
4 7
6 13
4 8
3 6
5 12
```

```
14
```

첫째 줄: N, K
나머지: 각 물건의 무게(W), 가치(V)

출력 예시