알고리즘 Week 7

20기 정규세션

TOBIG’S 19기 김은지

Contents

20기 정규세션
TOBIG'S 19기 김은지

20기 정규세션
TOBIG'S 19기 김은지

Unit 01 | 5주차 과제 리뷰

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제1. 피보나치 수 2 (2748번)

Dynamic Programming으로 풀이 가능한 대표적인 문제
중복된 하위 문제들을 갖기 때문!

문제

피보나치 수는 0과 1로 시작한다. 0번째 피보나치 수는 0이고, 1번째 피보나치 수는 1이다. 그 다음 2번째 부터는 바로 앞 두 피보나치 수의 합이 된다.

이를 식으로 써보면 $F_n = F_{n-1} + F_{n-2}$ $(n \geq 2)$가 된다.

n=17일때 까지 피보나치 수를 써보면 다음과 같다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597

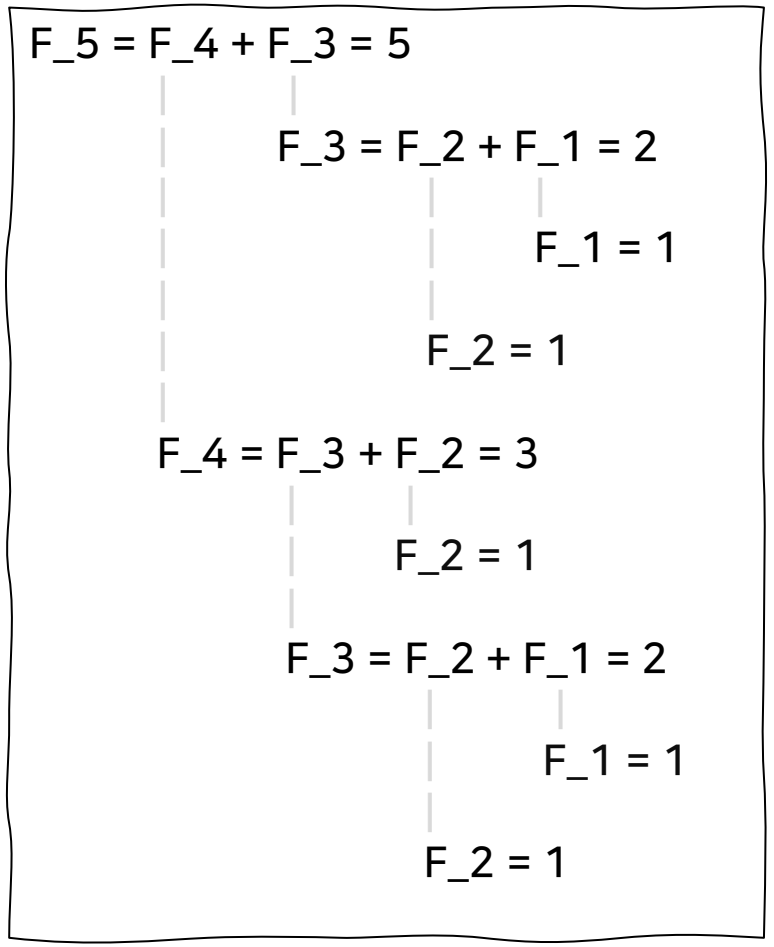
n이 주어졌을 때, n번째 피보나치 수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 n이 주어진다. n은 90보다 작거나 같은 자연수이다.

출력

첫째 줄에 n번째 피보나치 수를 출력한다.



Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제1. 피보나치 수 2 (2748번)

문제

피보나치 수는 0과 1로 시작한다. 0번째 피보나치 수는 0이고, 1번째 피보나치 수는 1이다. 그 다음 2번째 부터는 바로 앞 두 피보나치 수의 합이 된다.

이를 식으로 써보면 $F_n = F_{n-1} + F_{n-2}$ $(n \geq 2)$가 된다.

n=17일때 까지 피보나치 수를 써보면 다음과 같다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597

n이 주어졌을 때, n번째 피보나치 수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 n이 주어진다. n은 90보다 작거나 같은 자연수이다.

출력

첫째 줄에 n번째 피보나치 수를 출력한다.

Pseudo Code

```
Fib(n)
        if   n <= 1
                return   n
        else
                return   Fib(n-1) + Fib(n-2)
```

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제1. 피보나치 수 2 (2748번)

방법 1. 재귀 구조 브루트 포스

```python
def fib(N: int) -> int:
    if N <= 1:
        return N
    return fib(N-1) + fib(N-2)

n = int(input())

fibo = [fib(i) for i in range(n+1)]
print(fibo[n])
```

시간 초과!

방법 2. 메모이제이션(하향식)

```python
import collections

dp = collections.defaultdict(int)

def fib(N: int) -> int:
    if N <= 1:
        return N

    if dp[N]:
        return dp[N]
    dp[N] = fib(N-1) + fib(N-2)

    return dp[N]

n = int(input())
print(fib(n))
```

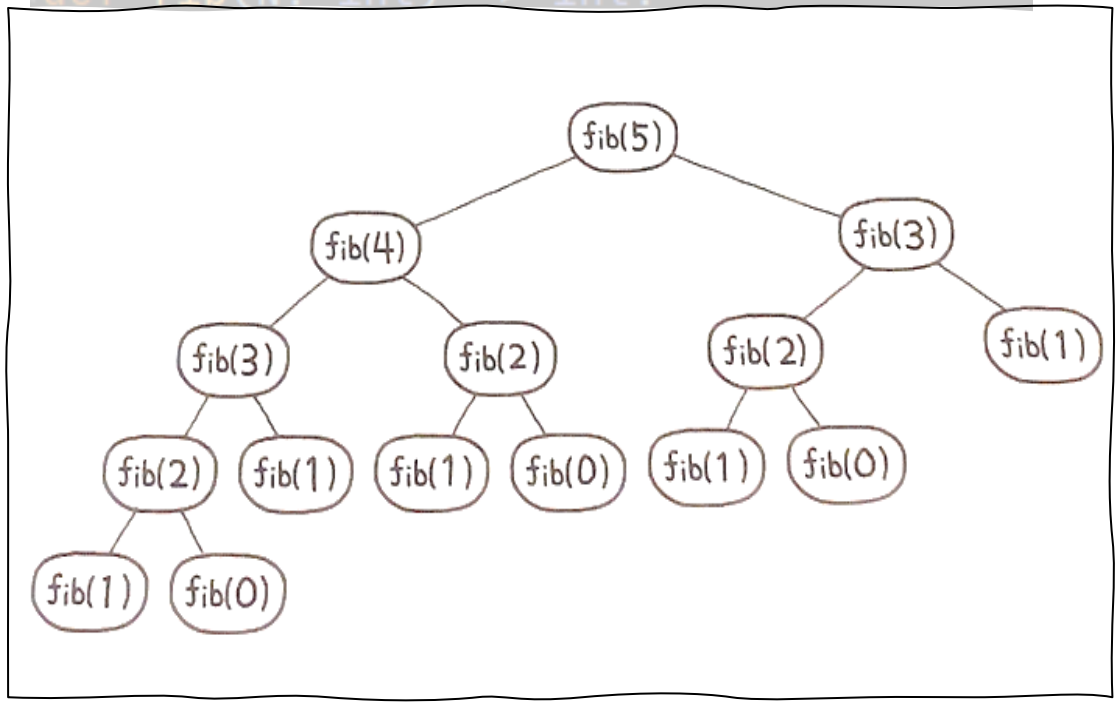
Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제1. 피보나치 수 2 (2748번)

방법 1. 재귀 구조 브루트 포스

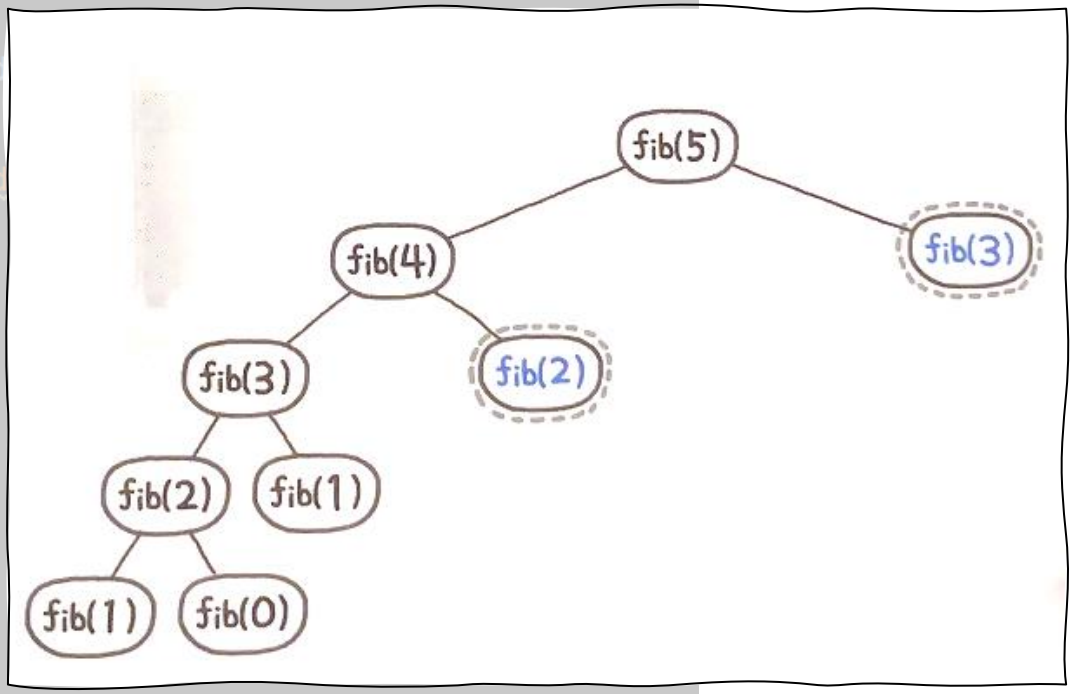
```
def fib(N: int) -> int:
```



15번의 연산

방법 2. 메모이제이션(하향식)

```
import collections
```



9번의 연산!

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제1. 피보나치 수 2 (2748번)

방법 3. 타뷸레이션(상향식)

```python
import collections

dp = collections.defaultdict(int)

def fib(N: int) -> int:
    dp[0] = 0
    dp[1] = 1

    for i in range(2, N+1):
        dp[i] = dp[i-1] + dp[i-2]

    return dp[N]

n = int(input())
print(fib(n))
```

방법 4. 두 변수만 이용해 공간 절약

```python
def fib(N: int) -> int:
    x, y = 0, 1

    for i in range(0, N):
        x, y = y, x + y

    return x

n = int(input())
print(fib(n))
```

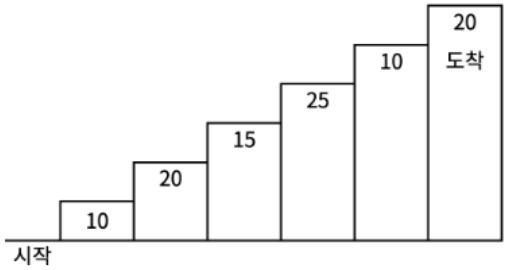
Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제2. 계단 오르기 (2579번)

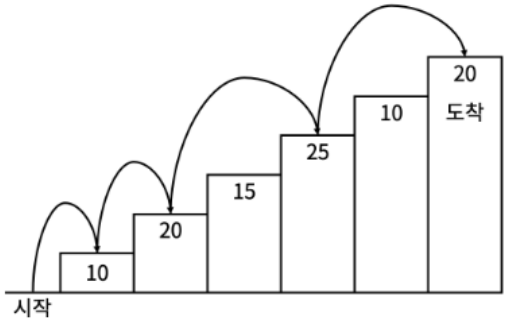
문제

계단 오르기 게임은 계단 아래 시작점부터 계단 꼭대기에 위치한 도착점까지 가는 게임이다. <그림 1>과 같이 각각의 계단에는 일정한 점수가 쓰여 있는데 계단을 밟으면 그 계단에 쓰여 있는 점수를 얻게 된다.



<그림 1>

예를 들어 <그림 2>와 같이 시작점에서부터 첫 번째, 두 번째, 네 번째, 여섯 번째 계단을 밟아 도착점에 도달하면 총 점수는 10 + 20 + 25 + 20 = 75점이 된다.



<그림 2>

계단 오르는 데는 다음과 같은 규칙이 있다.

1. 계단은 한 번에 한 계단씩 또는 두 계단씩 오를 수 있다. 즉, 한 계단을 밟으면서 이어서 다음 계단이나, 다음 다음 계단으로 오를 수 있다.
2. 연속된 세 개의 계단을 모두 밟아서는 안 된다. 단, 시작점은 계단에 포함되지 않는다.
3. 마지막 도착 계단은 반드시 밟아야 한다.

따라서 첫 번째 계단을 밟고 이어 두 번째 계단이나, 세 번째 계단으로 오를 수 있다. 하지만, 첫 번째 계단을 밟고 이어 네 번째 계단으로 올라가거나, 첫 번째, 두 번째, 세 번째 계단을 연속해서 모두 밟을 수는 없다.

각 계단에 쓰여 있는 점수가 주어질 때 이 게임에서 얻을 수 있는 총 점수의 최댓값을 구하는 프로그램을 작성하시오.

입력

입력의 첫째 줄에 계단의 개수가 주어진다.

둘째 줄부터 한 줄에 하나씩 제일 아래에 놓인 계단부터 순서대로 각 계단에 쓰여 있는 점수가 주어진다. 계단의 개수는 300이하의 자연수이고, 계단에 쓰여 있는 점수는 10,000이하의 자연수이다.

출력

첫째 줄에 계단 오르기 게임에서 얻을 수 있는 총 점수의 최댓값을 출력한다.

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제2. 계단 오르기 (2579번)

```python
n = int(input())  # 계단 개수
s = [int(input()) for _ in range(n)]  # 계단 리스트

dp= [0] * (n)  # dp 리스트

if len(s) <= 2:  # 계단이 2개 이하일땐 그냥 다 더해서 출력
    print(sum(s))

else:  # 계단이 3개 이상일 때
    dp[0]=s[0]  # 첫째 계단 수동 계산
    dp[1]=s[0]+s[1]  # 둘째 계단까지 수동 계산

    for i in range(2,n):  # 3번째 계단 부터 dp 점화식 이용해서 최대값 구하기
        dp[i] = max(dp[i-3]+s[i-1]+s[i], dp[i-2]+s[i])

    print(dp[-1])
```

둘 중 max값을 dp[i]에 할당

i-3계단까지의 점수 최댓값과
i-1, i계단 점수의 합

i-2계단까지의 점수 최댓값과
i계단 점수의 합

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

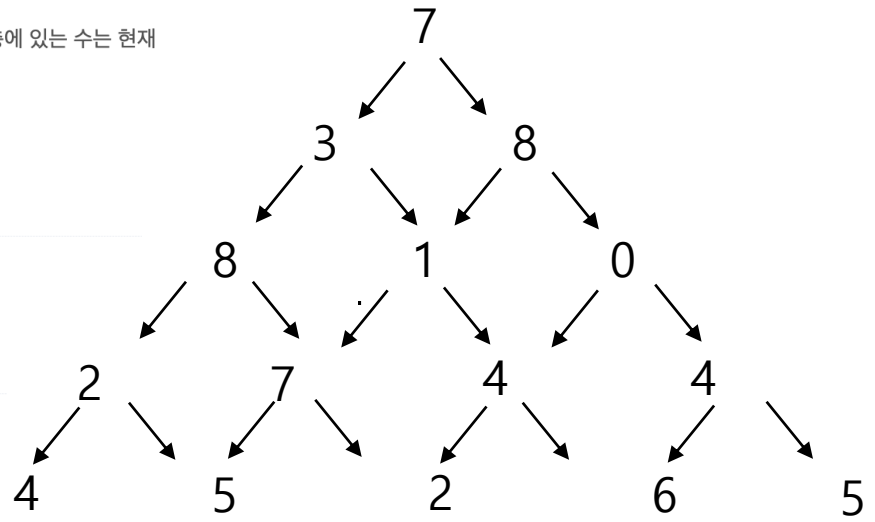
문제 문제3. 정수삼각형 (1932번)

```
        7
      3   8
    8   1   0
  2   7   4   4
4   5   2   6   5
```

위 그림은 크기가 5인 정수 삼각형의 한 모습이다.

맨 위층 7부터 시작해서 아래에 있는 수 중 하나를 선택하여 아래층으로 내려올 때, 이제까지 선택된 수의 합이 최대가 되는 경로를 구하는 프로그램을 작성하라. 아래층에 있는 수는 현재 층에서 선택된 수의 대각선 왼쪽 또는 대각선 오른쪽에 있는 것 중에서만 선택할 수 있다.

삼각형의 크기는 1 이상 500 이하이다. 삼각형을 이루고 있는 각 수는 모두 정수이며, 범위는 0 이상 9999 이하이다.



입력

첫째 줄에 삼각형의 크기 n(1 ≤ n ≤ 500)이 주어지고, 둘째 줄부터 n+1번째 줄까지 정수 삼각형이 주어진다.

출력

첫째 줄에 합이 최대가 되는 경로에 있는 수의 합을 출력한다.

예제 입력 1 복사

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

예제 출력 1 복사

```
30
```

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

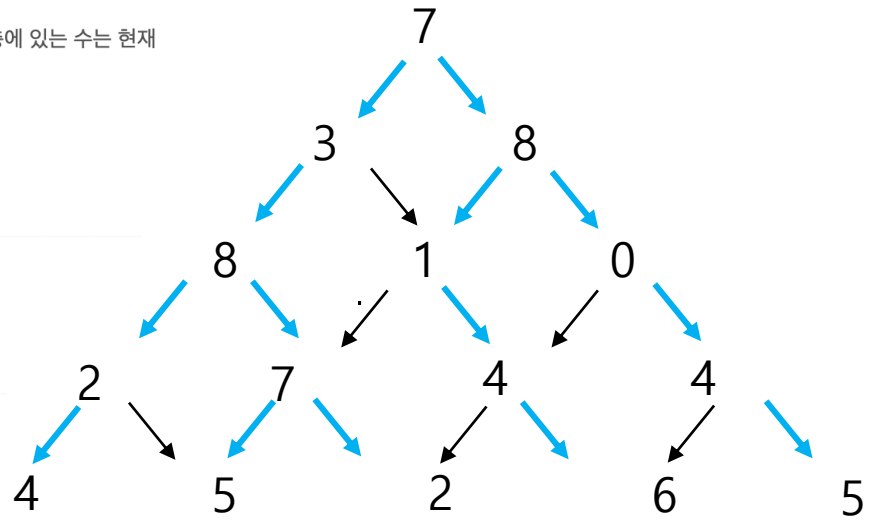
문제 문제3. 정수삼각형 (1932번)

```
        7
      3   8
    8   1   0
  2   7   4   4
4   5   2   6   5
```

위 그림은 크기가 5인 정수 삼각형의 한 모습이다.

맨 위층 7부터 시작해서 아래에 있는 수 중 하나를 선택하여 아래층으로 내려올 때, 이제까지 선택된 수의 합이 최대가 되는 경로를 구하는 프로그램을 작성하라. 아래층에 있는 수는 현재 층에서 선택된 수의 대각선 왼쪽 또는 대각선 오른쪽에 있는 것 중에서만 선택할 수 있다.

삼각형의 크기는 1 이상 500 이하이다. 삼각형을 이루고 있는 각 수는 모두 정수이며, 범위는 0 이상 9999 이하이다.



입력

첫째 줄에 삼각형의 크기 n(1 ≤ n ≤ 500)이 주어지고, 둘째 줄부터 n+1번째 줄까지 정수 삼각형이 주어진다.

출력

첫째 줄에 합이 최대가 되는 경로에 있는 수의 합을 출력한다.

예제 입력 1 복사

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

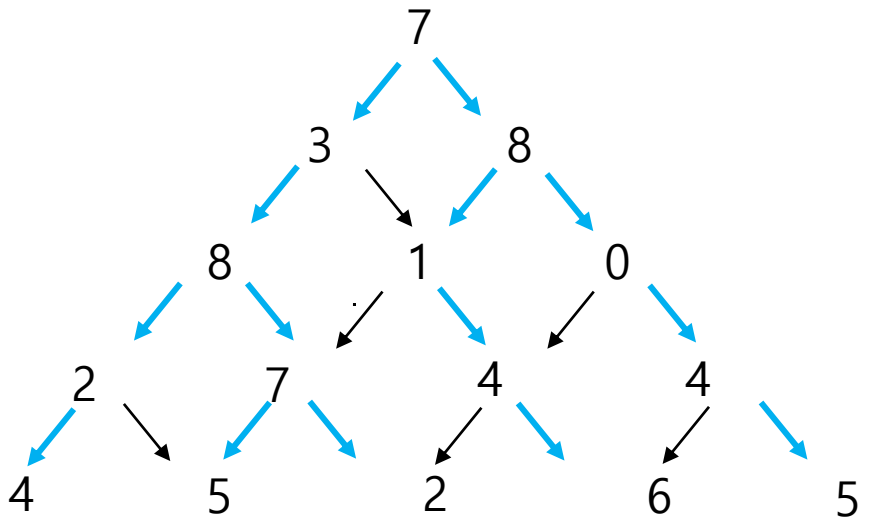
예제 출력 1 복사

```
30
```

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제3. 정수삼각형 (1932번)




```python
n = int(input())
dp = []

for i in range(n):
    dp.append(list(map(int, input().split())))


for i in range(1, n):
    for j in range(0, i+1):
        if j == 0:  # 각 라인의 처음은
            dp[i][0] += dp[i-1][0]  # 바로 위의 숫자를 더해주기
        elif j == i:    # 각 라인의 마지막은
            dp[i][j] += dp[i-1][j-1]    # 바로 위의 숫자를 더해주기
        else:   # 각 라인의 처음과 끝이 아닌 경우에는
            dp[i][j] += max(dp[i-1][j-1], dp[i-1][j])   # 왼쪽 or 오른쪽 대각선 중 최대값을 더하기

print(max(dp[n-1])) # n-1행(가장 마지막 행)에서의 최댓값 출력
```

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제4. 퇴사 (14501번)

상담원으로 일하고 있는 백준이는 퇴사를 하려고 한다.

오늘부터 N+1일째 되는 날 퇴사를 하기 위해서 남은 N일 동안 최대한 많은 상담을 하려고 한다.

백준이는 비서에게 최대한 많은 상담을 잡으라고 부탁을 했고, 비서는 하루에 하나씩 서로 다른 사람의 상담을 잡아놓았다.

각각의 상담은 상담을 완료하는데 걸리는 기간 T_i와 상담을 했을 때 받을 수 있는 금액 P_i로 이루어져 있다.

N = 7인 경우에 다음과 같은 상담 일정표를 보자.

	1일	2일	3일	4일	5일	6일	7일
T_i	3	5	1	1	2	4	2
P_i	10	20	10	20	15	40	200

1일에 잡혀있는 상담은 총 3일이 걸리며, 상담했을 때 받을 수 있는 금액은 10이다. 5일에 잡혀있는 상담은 총 2일이 걸리며, 받을 수 있는 금액은 15이다.

상담을 하는데 필요한 기간은 1일보다 클 수 있기 때문에, 모든 상담을 할 수는 없다. 예를 들어서 1일에 상담을 하게 되면, 2일, 3일에 있는 상담은 할 수 없게 된다. 2일에 있는 상담을 하게 되면, 3, 4, 5, 6일에 잡혀있는 상담은 할 수 없다.

또한, N+1일째에는 회사에 없기 때문에, 6, 7일에 있는 상담을 할 수 없다.

퇴사 전에 할 수 있는 상담의 최대 이익은 1일, 4일, 5일에 있는 상담을 하는 것이며, 이때의 이익은 10+20+15=45이다.

상담을 적절히 했을 때, 백준이가 얻을 수 있는 최대 수익을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N ($1 \le N \le 15$)이 주어진다.

둘째 줄부터 N개의 줄에 T_i와 P_i가 공백으로 구분되어서 주어지며, 1일부터 N일까지 순서대로 주어진다. ($1 \le T_i \le 5$, $1 \le P_i \le 1,000$)

출력

첫째 줄에 백준이가 얻을 수 있는 최대 이익을 출력한다.

예제 입력 1 복사

```
7
3 10
5 20
1 10
1 20
2 15
4 40
2 200
```

예제 출력 1 복사

```
45
```

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제4. 퇴사 (14501번)

방법 1. 타뷸레이션(상향식)

```python
N = int(input())

schdl = [list(map(int, input().split())) for _ in range(N)]

dp = [0 for _ in range(N+1)]

for i in range(N):
    for j in range(i + schdl[i][0], N+1):
        if dp[j] < dp[i] + schdl[i][1]:
            dp[j] = dp[i] + schdl[i][1]

print(dp[-1])
```

방법 2. 메모이제이션(하향식)

```python
N = int(input())

schdl = [list(map(int, input().split())) for _ in range(N)]

dp = [0 for _ in range(N+1)]

for i in range(N-1, -1, -1):
    if i + schdl[i][0] > N:
        dp[i] = dp[i+1]
    else:
        dp[i] = max(dp[i+1], schdl[i][1] + dp[i + schdl[i][0]])

print(dp[0])
```

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제5. 동전1 (2293번)

문제

n가지 종류의 동전이 있다. 각각의 동전이 나타내는 가치는 다르다. 이 동전을 적당히 사용해서, 그 가치의 합이 k원이 되도록 하고 싶다. 그 경우의 수를 구하시오. 각각의 동전은 몇 개라도 사용할 수 있다.

사용한 동전의 구성이 같은데, 순서만 다른 것은 같은 경우이다.

입력

첫째 줄에 n, k가 주어진다. ($1 \le n \le 100$, $1 \le k \le 10,000$) 다음 n개의 줄에는 각각의 동전의 가치가 주어진다. 동전의 가치는 100,000보다 작거나 같은 자연수이다.

출력

첫째 줄에 경우의 수를 출력한다. 경우의 수는 2^{31}보다 작다.

예제 입력 1 복사

```
3 10
1
2
5
```

예제 출력 1 복사

```
10
```

Unit 01 | 5주차 과제 리뷰

20기 정규세션
TOBIG'S 19기 김은지

문제5. 동전1 (2293번)

```python
# 1. n, k 입력받기
n, k = map(int, input().split())

# 2. coin 종류 입력 받기
coins = []
for i in range(n):
    coins.append(int(input()))

# 3. dp 리스트 만들기
dp = [0] * (k+1)
dp[0] = 1

# 4. for문 통해서 경우의 수 누적 합 구하기
for c in coins: # coin 종류를 돌면서
    for j in range(c, k+1): # 앞에 탐색한 경우는 제외하고, 주어진 합에서 동전을 뺀 수가 양수이면
        # 즉, coin을 더 이용해서 합을 만들어야 할 경우이면
        p_case = dp[j - c]
        dp[j] += p_case

print(dp[k])    # 구하고자 하는 합의 경우의 수 출력하기
```

각 동전으로 만들 수 있는 금액을 계산

이전의 동전으로 진행했던 부분 +
새로운 동전으로 만들 수 있는 방법

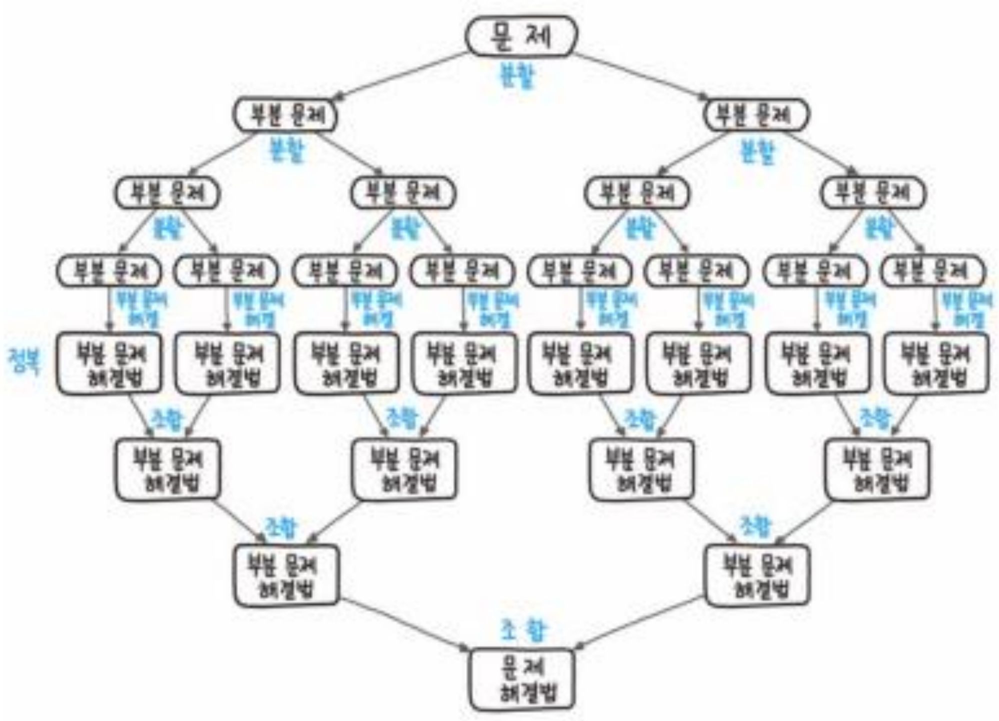
이전에 동전으로 만들었던 부분을
새로운 코인으로 만들 수 있는 만큼 추가

20기 정규세션
TOBIG'S 19기 김은지

Unit 02 | 정렬 및 분할정복

분할정복(Divide and Conquer)

주어진 문제가 간단한 문제가 될 때까지 문제를 재귀적으로 나눈 다음
각 문제의 결과를 조합하여 전체 문제의 답을 계산하는 방식



Unit 02 | 정렬 및 분할정복

20기 정규세션
TOBIG'S 19기 김은지

분할정복의 과정

1. **분할** : 문제를 동일한 유형의 여러 하위 문제로 나눈다.

2. **정복** : 가장 작은 단위의 하위 문제를 해결하여 정복한다.

3. **조합** : 하위 문제에 대한 결과를 원래 문제에 대한 결과로 조합한다.

Unit 02 | 정렬 및 분할정복

20기 정규세션
TOBIG'S 19기 김은지

분할정복의 장점과 단점

장점

문제를 나눔으로써
어려운 문제를 해결할 수 있게 된다

단점

함수를 재귀적으로 호출한다는 점에서
함수 호출로 인한 오버헤드가 발생,

스택에 다양한 데이터를 보관하고
있어야 하므로 스택 오버플로우가 발생하거나
과도한 메모리 사용을 하게 된다.

※오버헤드: 어떤 처리를 하기 위해 들어가는 간접적인 처리 시간/메모리
※스택 오버플로우:변수의 크기가 스택보다 크거나, 함수를 무한으로 호출하고 있을 때, 혹은 스택을 넘어가 다른 곳에 위치하는 경우 발생

20기 정규세션
TOBIG'S 19기 김은지

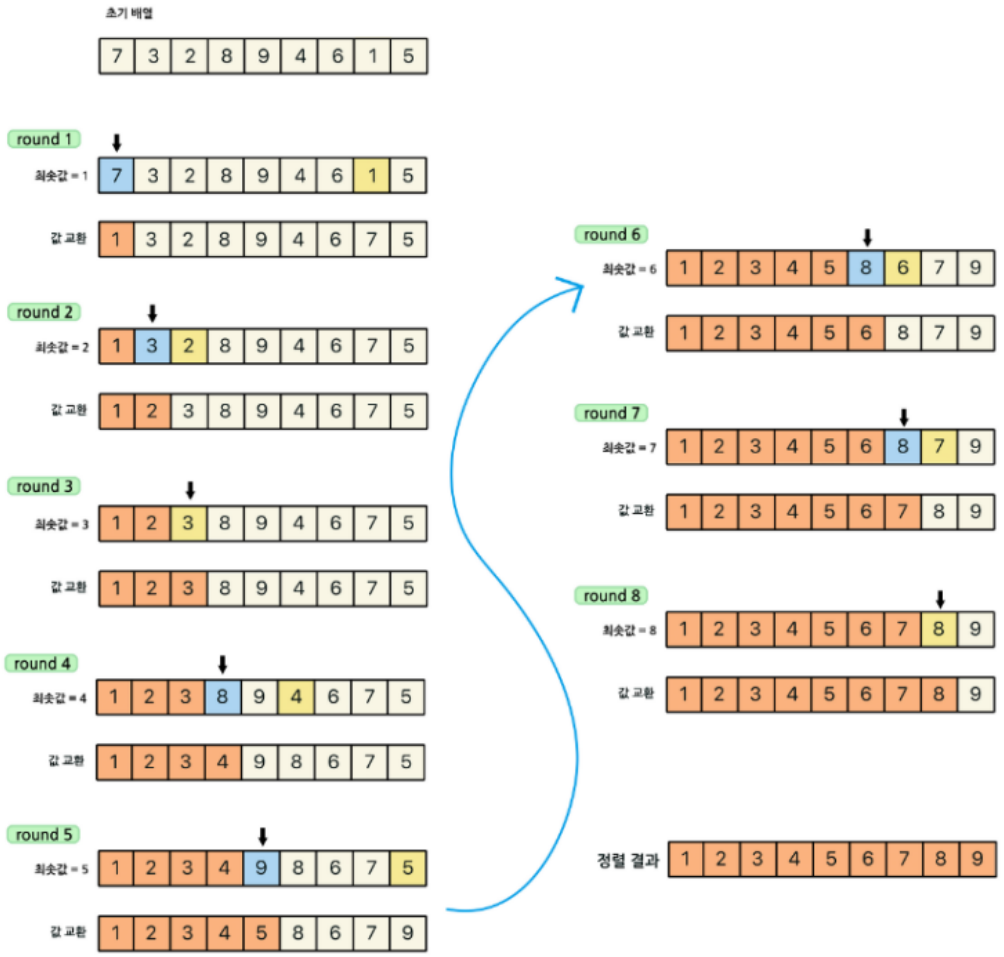
Unit 03 | 정렬 알고리즘 소개

Unit 03 | 정렬 알고리즘 소개

20기 정규세션
TOBIG'S 19기 김은지

선택정렬

정렬이 되지 않은 숫자들 중에서 최솟값을 선택하여
배열의 첫번째 요소와 교환하는 정렬 알고리즘



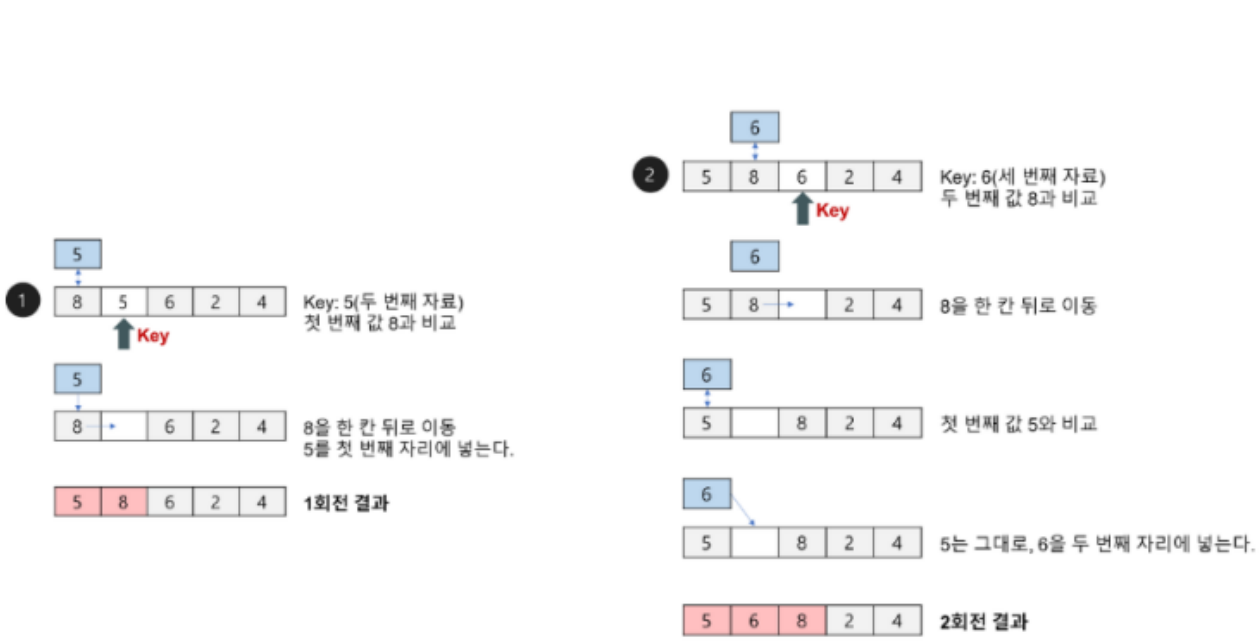

```python
def selection_sort(arr):
    for i in range(len(arr) - 1):
        min_idx = i
        for j in range(i+1, len(arr)):
            if arr[min_idx] > arr[j]:
                min_idx = j

        arr[i], arr[min_idx] = arr[min_idx], arr[i]
    return arr
```

Unit 03 | 정렬 알고리즘 소개

20기 정규세션
TOBIG'S 19기 김은지

삽입정렬

두번째 원소부터 시작해 그 앞에 존재하는 원소들과 비교하여
삽입할 위치를 찾아 삽입하는 정렬 알고리즘



Unit 03 | 정렬 알고리즘 소개

20기 정규세션
TOBIG'S 19기 김은지

삽입정렬

두번째 원소부터 시작해 그 앞에 존재하는 원소들과 비교하여
삽입할 위치를 찾아 삽입하는 정렬 알고리즘

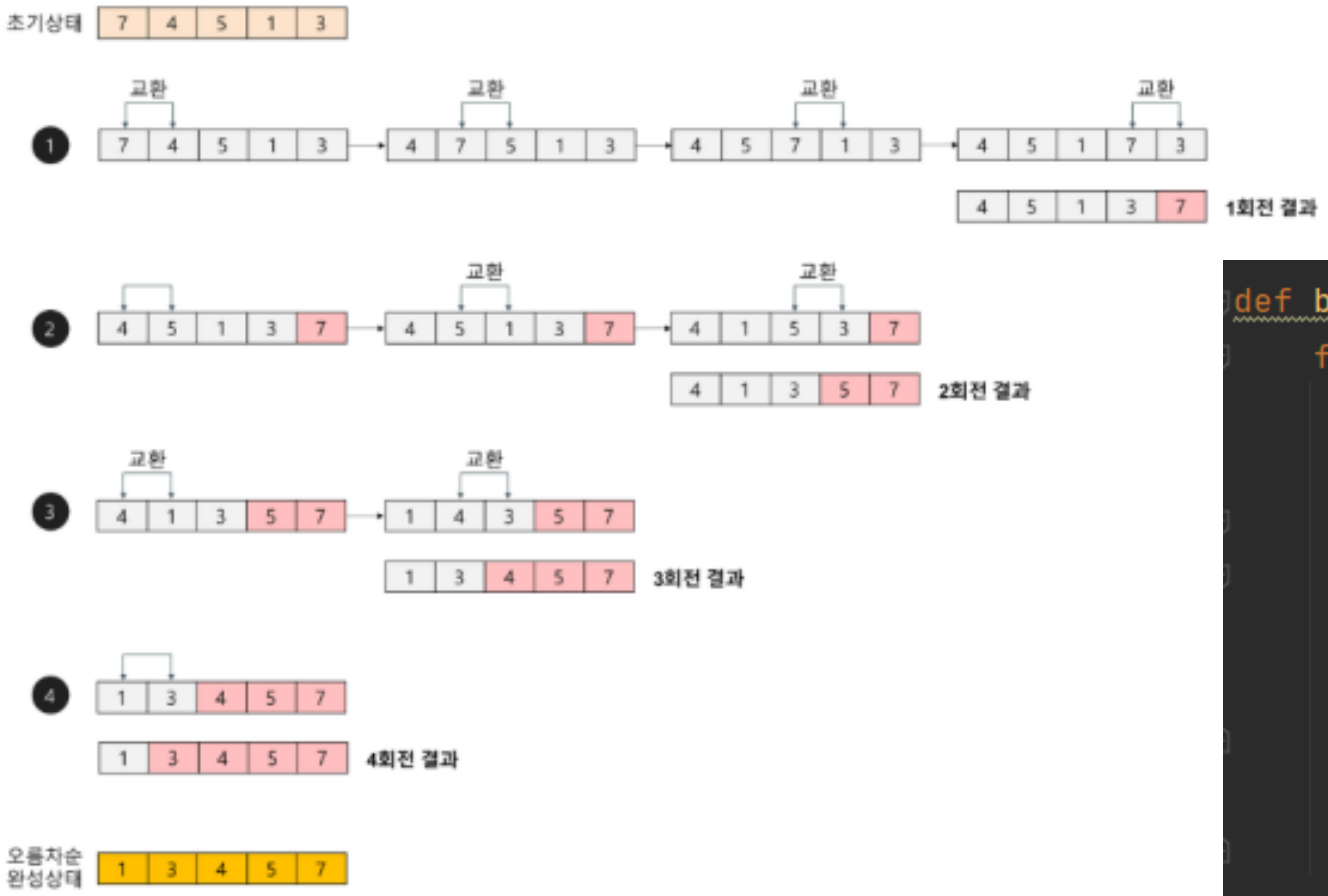
```python
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key_item = arr[i]
        j = i-1

        while j >= 0 and arr[j] > key_item:
            arr[j+1] = arr[j]
            j -= 1

        arr[j+1] = key_item

    return arr
```

Unit 03 | 정렬 알고리즘 소개

20기 정규세션
TOBIG'S 19기 김은지

버블정렬

서로 인접한 두 원소를 비교하여 정렬하는 알고리즘



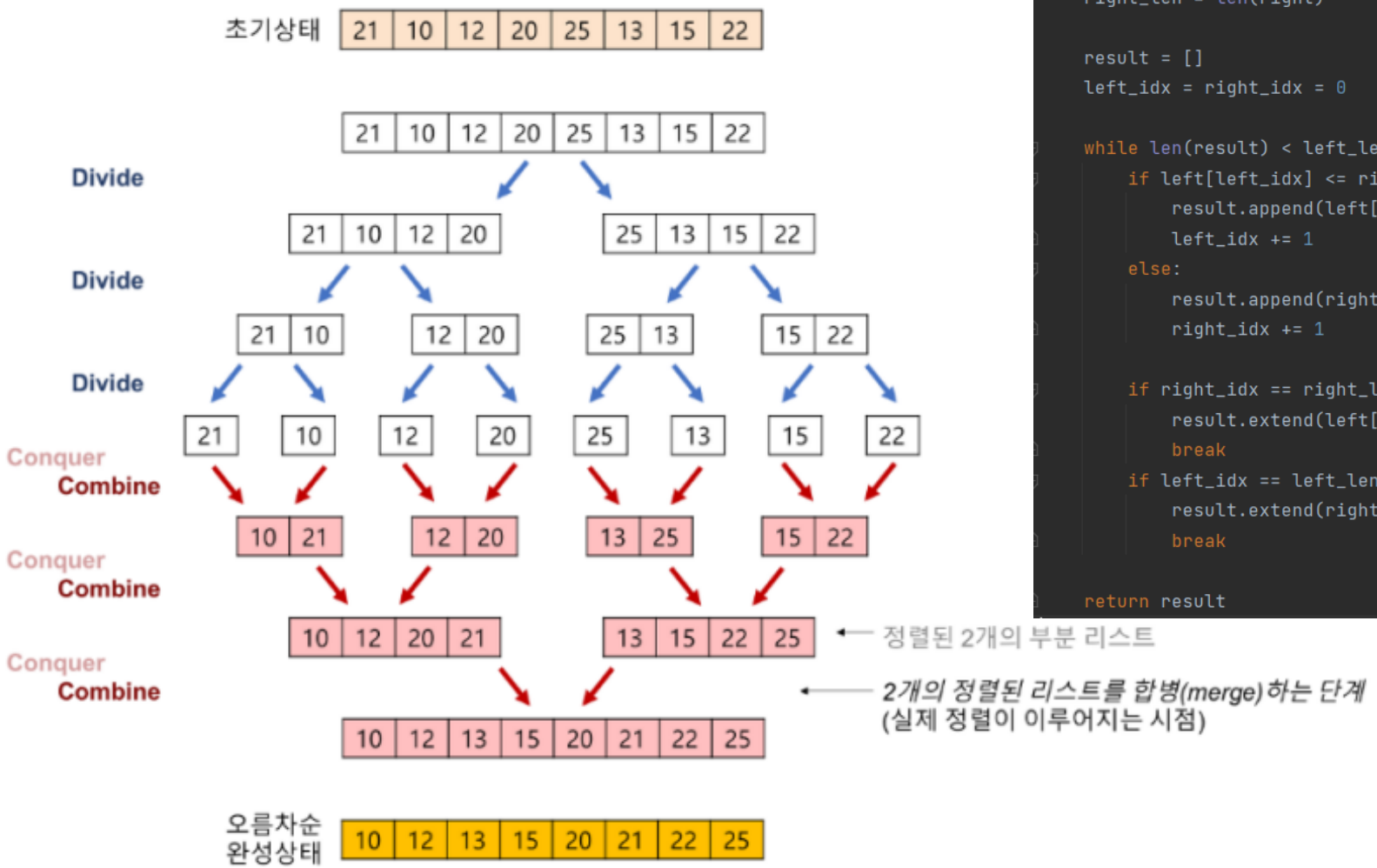

```python
def bubble_sort(arr):
    for i in range(len(arr) - 1):
        done_sort = True

        for j in range(len(arr) - i - 1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

                done_sort = True
        if done_sort:
            break

    return arr
```

Unit 03 | 정렬 알고리즘 소개

20기 정규세션
TOBIG'S 19기 김은지

Merge Sort

주어진 배열을 크기가 1인 배열로 분할하고 합병하면서 정렬을 진행하는 분할정복 알고리즘




```python
def merge(left, right):
    left_len = len(left)
    right_len = len(right)

    result = []
    left_idx = right_idx = 0

    while len(result) < left_len + right_len:
        if left[left_idx] <= right[right_idx]:
            result.append(left[left_idx])
            left_idx += 1
        else:
            result.append(right[right_idx])
            right_idx += 1

        if right_idx == right_len:
            result.extend(left[left_idx:])
            break
        if left_idx == left_len:
            result.extend(right[right_idx:])
            break

    return result
```

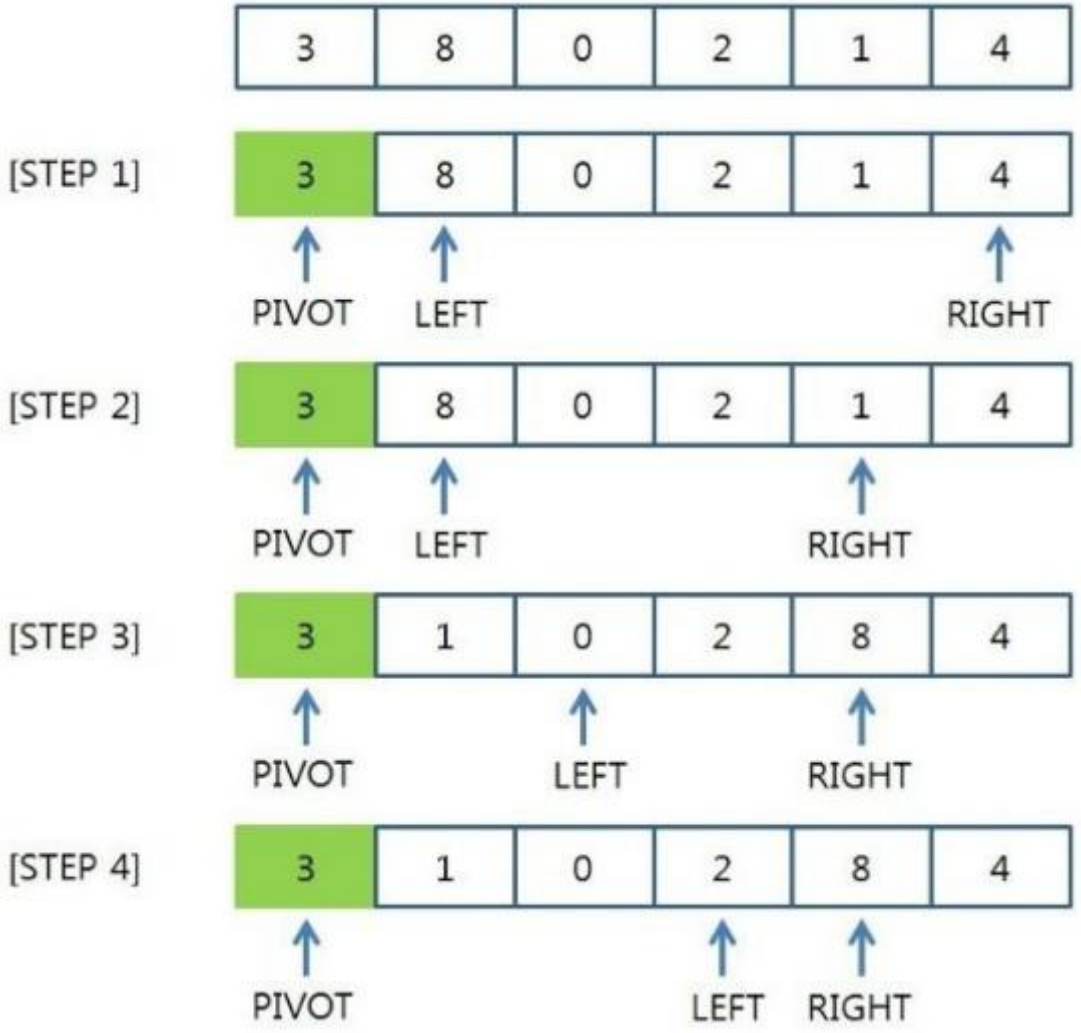
```python
def merge_sort(arr):
    if len(arr) < 2:
        return arr

    mid_idx = len(arr) // 2
    left = merge_sort(arr[:mid_idx])
    right = merge_sort(arr[mid_idx:])

    return merge(left, right)
```

Unit 03 | 정렬 알고리즘 소개

20기 정규세션
TOBIG'S 19기 김은지

Quick Sort

Merge Sort와 다르게 리스트를 비균등하게 분할
Pivot을 설정하고 Pivot보다 큰 값, 작은 값으로 분할하여 정렬




```python
def quicksort(arr):
    if len(arr) < 2:
        return arr

    low, same, high = [], [], []
    pivot = arr[randint(0, len(arr)-1)]

    for item in arr:
        if item < pivot:
            low.append(item)
        elif item == pivot:
            same.append(item)
        elif item > pivot:
            high.append(item)

    return quicksort(low) + same + quicksort(high)
```

Unit 04 | 7주차 과제 소개

Unit 04 | 7주차 과제 소개

20기 정규세션
TOBIG'S 19기 김은지

문제 1. 좌표 정렬하기 https://www.acmicpc.net/problem/11650

문제

2차원 평면 위의 점 N개가 주어진다. 좌표를 x좌표가 증가하는 순으로, x좌표가 같으면 y좌표가 증가하는 순서로 정렬한 다음 출력하는 프로그램을 작성하시오.

입력

첫째 줄에 점의 개수 N ($1 \le N \le 100,000$)이 주어진다. 둘째 줄부터 N개의 줄에는 i번점의 위치 x_i와 y_i가 주어진다. ($-100,000 \le x_i, y_i \le 100,000$) 좌표는 항상 정수이고, 위치가 같은 두 점은 없다.

출력

첫째 줄부터 N개의 줄에 점을 정렬한 결과를 출력한다.

예제 입력 1 복사

```
5
3 4
1 1
1 -1
2 2
3 3
```

예제 출력 1 복사

```
1 -1
1 1
2 2
3 3
3 4
```

Unit 04 | 7주차 과제 소개

20기 정규세션
TOBIG'S 19기 김은지

문제 2. 색종이 만들기

https://www.acmicpc.net/problem/2630

문제

아래 <그림 1>과 같이 여러개의 정사각형칸들로 이루어진 정사각형 모양의 종이가 주어져 있고, 각 정사각형들은 하얀색으로 칠해져 있거나 파란색으로 칠해져 있다. 주어진 종이를 일정한 규칙에 따라 잘라서 다양한 크기를 가진 정사각형 모양의 하얀색 또는 파란색 색종이를 만들려고 한다.

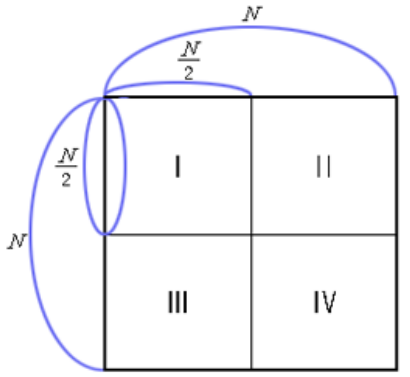
1	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
1	0	0	0	1	1	1	1
0	1	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1

<그림 1> 8×8 종이

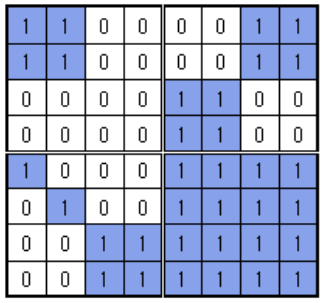
전체 종이의 크기가 N×N(N=2^k, k는 1 이상 7 이하의 자연수) 이라면 종이를 자르는 규칙은 다음과 같다.

전체 종이가 모두 같은 색으로 칠해져 있지 않으면 가로와 세로로 중간 부분을 잘라서 <그림 2>의 I, II, III, IV와 같이 똑같은 크기의 네 개의 N/2 × N/2색종이로 나눈다. 나누어진 종이 I, II, III, IV 각각에 대해서도 앞에서와 마찬가지로 모두 같은 색으로 칠해져 있지 않으면 같은 방법으로 똑같은 크기의 네 개의 색종이로 나눈다. 이와 같은 과정을 잘라진 종이가 모두 하얀색 또는 모두 파란색으로 칠해져 있거나, 하나의 정사각형 칸이 되어 더 이상 자를 수 없을 때까지 반복한다.

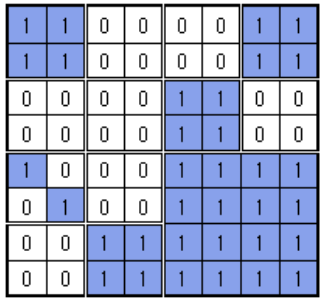
위와 같은 규칙에 따라 잘랐을 때 <그림 3>은 <그림 1>의 종이를 처음 나눈 후의 상태를, <그림 4>는 두 번째 나눈 후의 상태를, <그림 5>는 최종적으로 만들어진 다양한 크기의 9장의 하얀색 색종이와 7장의 파란색 색종이를 보여주고 있다.



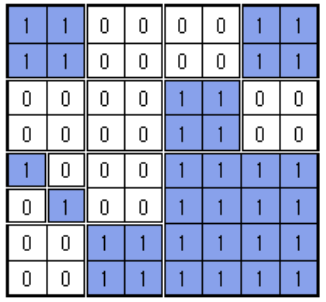
<그림 2>



<그림 3> 처음 나눈 후의 상태



<그림 4> 두 번째 나눈 후의 상태



<그림 5> 최종적으로 나누어진 색종이들

입력으로 주어진 종이의 한 변의 길이 N과 각 정사각형칸의 색(하얀색 또는 파란색)이 주어질 때 잘라진 하얀색 색종이와 파란색 색종이의 개수를 구하는 프로그램을 작성하시오.

Unit 04 | 7주차 과제 소개

20기 정규세션
TOBIG'S 19기 김은지

문제 2. 색종이 만들기

https://www.acmicpc.net/problem/2630

입력

첫째 줄에는 전체 종이의 한 변의 길이 N이 주어져 있다. N은 2, 4, 8, 16, 32, 64, 128 중 하나이다. 색종이의 각 가로줄의 정사각형칸들의 색이 윗줄부터 차례로 둘째 줄부터 마지막 줄까지 주어진다. 하얀색으로 칠해진 칸은 0, 파란색으로 칠해진 칸은 1로 주어지며, 각 숫자 사이에는 빈칸이 하나씩 있다.

출력

첫째 줄에는 잘라진 하얀색 색종이의 개수를 출력하고, 둘째 줄에는 파란색 색종이의 개수를 출력한다.

예제 입력 1 복사

```
8
1 1 0 0 0 0 1 1
1 1 0 0 0 0 1 1
0 0 0 0 1 1 0 0
0 0 0 0 1 1 0 0
1 0 0 0 1 1 1 1
0 1 0 0 1 1 1 1
0 0 1 1 1 1 1 1
0 0 1 1 1 1 1 1
```

예제 출력 1 복사

```
9
7
```

Unit 04 | 7주차 과제 소개

20기 정규세션
TOBIG'S 19기 김은지

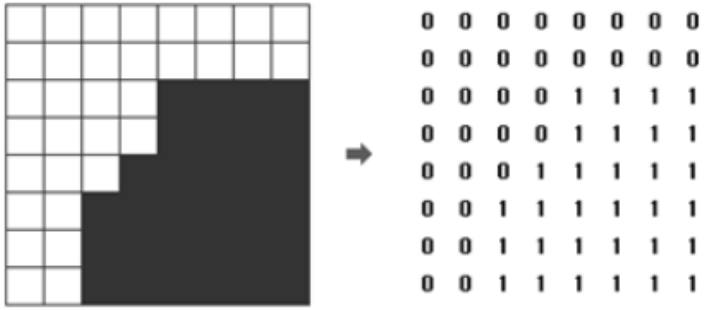
문제 3. 쿼드트리

https://www.acmicpc.net/problem/1992

문제

흑백 영상을 압축하여 표현하는 데이터 구조로 쿼드 트리(Quad Tree)라는 방법이 있다. 흰 점을 나타내는 0과 검은 점을 나타내는 1로만 이루어진 영상(2차원 배열)에서 같은 숫자의 점들이 한 곳에 많이 몰려있으면, 쿼드 트리에서는 이를 압축하여 간단히 표현할 수 있다.

주어진 영상이 모두 0으로만 되어 있으면 압축 결과는 "0"이 되고, 모두 1로만 되어 있으면 압축 결과는 "1"이 된다. 만약 0과 1이 섞여 있으면 전체를 한 번에 나타내지를 못하고, 왼쪽 위, 오른쪽 위, 왼쪽 아래, 오른쪽 아래, 이렇게 4개의 영상으로 나누어 압축하게 되며, 이 4개의 영역을 압축한 결과를 차례대로 괄호 안에 묶어서 표현한다



```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1
0 0 0 1 1 1 1 1
0 0 1 1 1 1 1 1
0 0 1 1 1 1 1 1
0 0 1 1 1 1 1 1
```

위 그림에서 왼쪽의 영상은 오른쪽의 배열과 같이 숫자로 주어지며, 이 영상을 쿼드 트리 구조를 이용하여 압축하면 "`(0(0011)(0(0111)01)1)`"로 표현된다. N ×N 크기의 영상이 주어질 때, 이 영상을 압축한 결과를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에는 영상의 크기를 나타내는 숫자 N 이 주어진다. N 은 언제나 2의 제곱수로 주어지며, 1 ≤ N ≤ 64의 범위를 가진다. 두 번째 줄부터는 길이 N의 문자열이 N개 들어온다. 각 문자열은 0 또는 1의 숫자로 이루어져 있으며, 영상의 각 점들을 나타낸다.

출력

영상을 압축한 결과를 출력한다.

예제 입력 1 복사

```
8
11110000
11110000
00011100
00011100
11110000
11110000
11110011
11110011
```

예제 출력 1 복사

```
((110(0101))(0010)1(0001))
```

Unit 04 | 7주차 과제 소개

20기 정규세션
TOBIG'S 19기 김은지

7주차 알고리즘 과제 기한 : 9월 13일 20:00 (2주)

질문/힌트/모든 문의는
19기 김은지/ 위성진/ 이동준/ 임서영 최다희/ 표지원/ 한진모 에게

언제든 자유롭게 해주세요!

20기 정규세션
TOBIG'S 19기 김은지

알고리즘 화이팅