

02강

알고리즘과 자료구조

Brute-force 전략1

서울과학기술대학교 신일훈 교수

학습목표

- 1 brute-force 전략의 개념을 이해한다.
- 2 합계를 구하는 알고리즘을 설계할 수 있다.
- 3 최대값, 최소값을 구하는 알고리즘을 설계할 수 있다.
- 4 약수의 개수를 구하는 알고리즘을 설계할 수 있다.





brute-force 전략 개념

1. brute-force 전략 개념

■ brute-force (억지) 전략 개념

- 답을 찾기 위해, 모든 가능한 경우를 전부 확인하거나 수행하는 단순한 방법
 - 주로 반복 기법 활용
 - 컴퓨터의 빠른 성능을 활용

1. brute-force 전략 개념

■ brute-force (억지) 전략 예시

- 1-100까지의 합계를 구하는 방법
 - $\text{sum} = 1 + 2 + \dots + 100$
 - 반복문으로 구현
- 9^n 을 구하는 방법
 - $\text{result} = 9 * 9 * \dots * 9$

1. brute-force 전략 개념

■ brute-force (억지) 전략 예시

- 오름차순으로 정렬된 숫자들의 리스트에서 최대값 찾는 방법
 - 첫번째 값을 max로 초기화
 - 리스트의 두번째부터 마지막 원소를 현재의 max와 비교하여, max가 작으면, max를 리스트의 원소값으로 변경.
- 서울에서 출발하여, 대전, 부산, 광주를 모두 방문하고 다시 서울로 돌아오는 최단경로는? (방문 순서는 상관없음)
 - 경로1: 서울 - 대전 - 부산 - 광주 - 서울
 - 경로2: 서울 - 대전 - 광주 - 부산 - 서울
 - ...

} 최소값 경로 선택

1. brute-force 전략 개념

■ brute-force (억지) 전략 특성

- easy to design
- 광범위한 문제에 적용 가능
- 입력(데이터)의 크기(개수)가 작은 경우 효과적
- 입력(데이터)의 크기(개수)가 큰 경우 시간 복잡도가 높을 수 있음
 - 낮은 성능
- 더 효율적인 알고리즘을 통해 성능 개선 가능

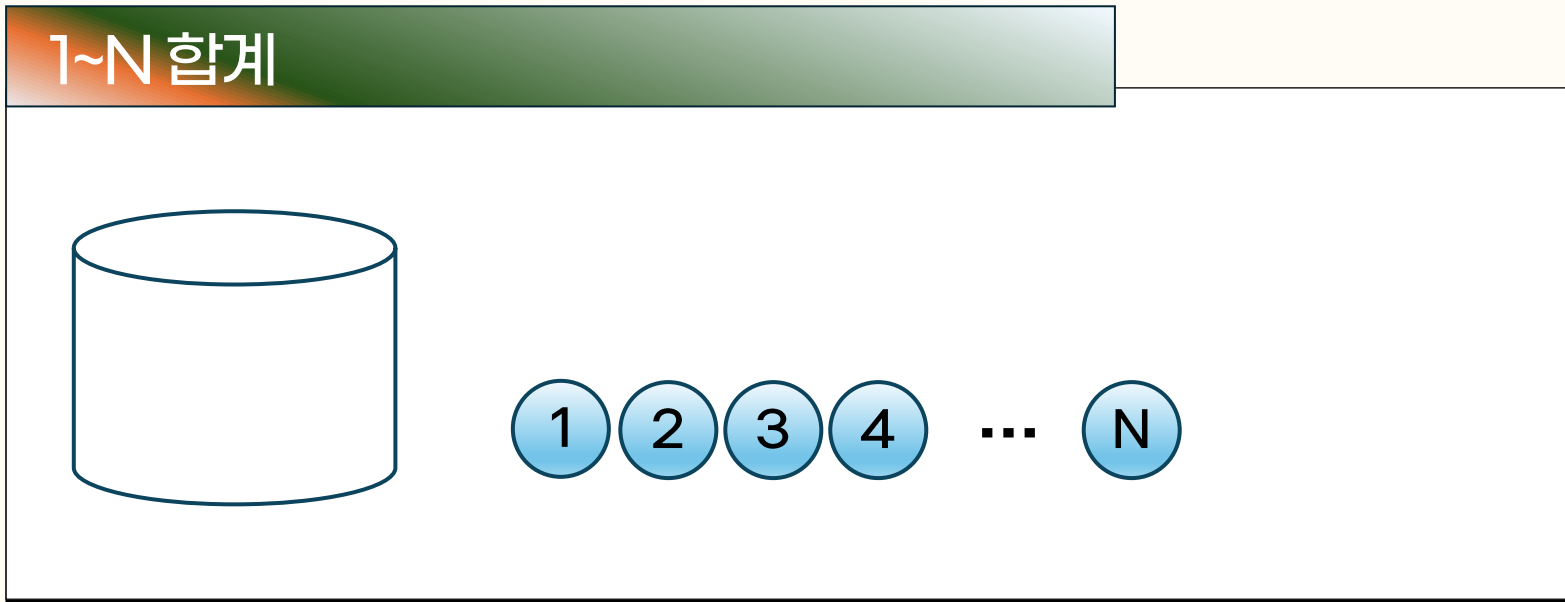


합계구하기

2. 합계 구하기

■ 1부터 자연수 N 까지의 합계를 구하시오.

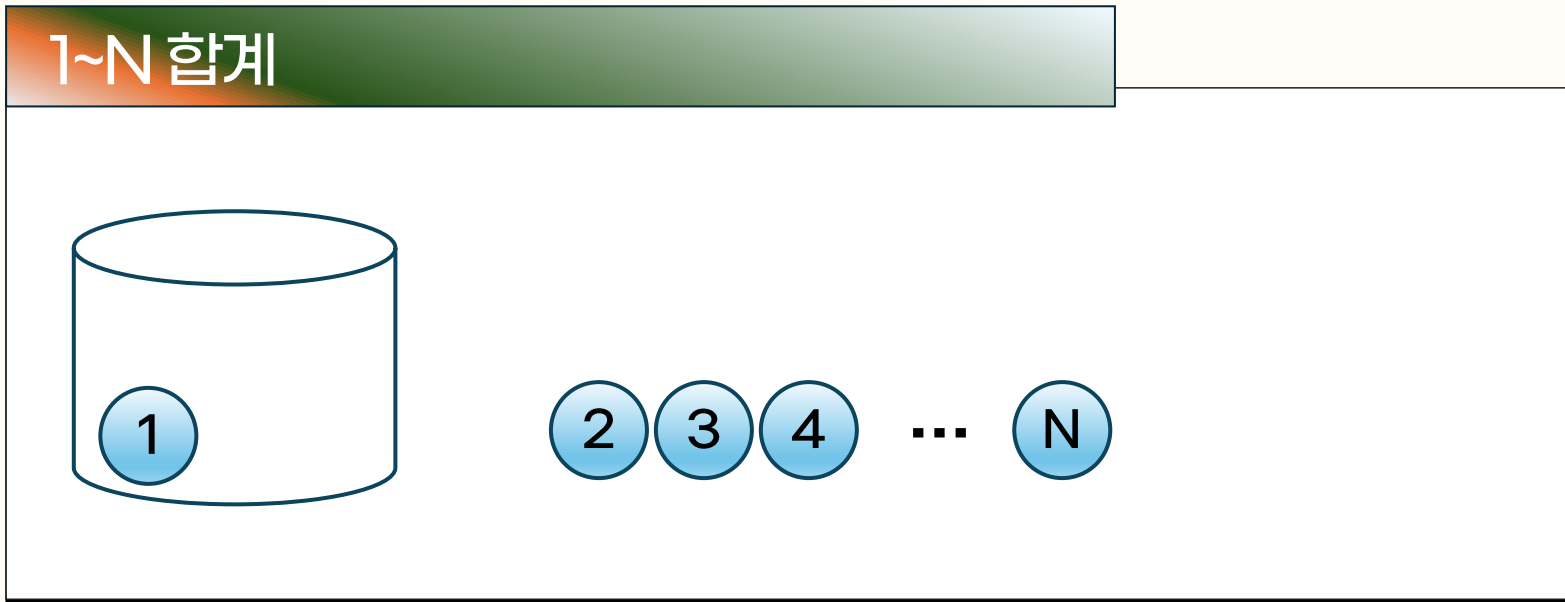
- 알고리즘?



2. 합계 구하기

■ 1부터 자연수 N까지의 합계를 구하시오.

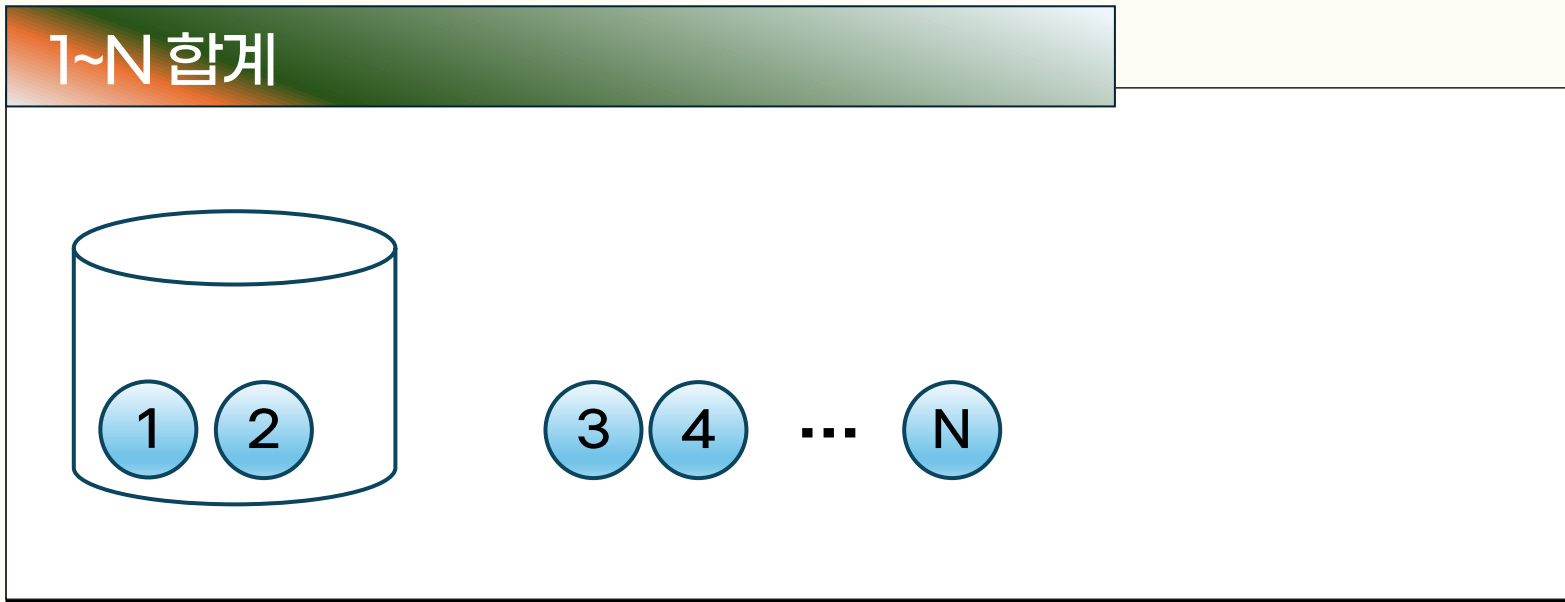
- 알고리즘?



2. 합계 구하기

■ 1부터 자연수 N 까지의 합계를 구하시오.

- 알고리즘?

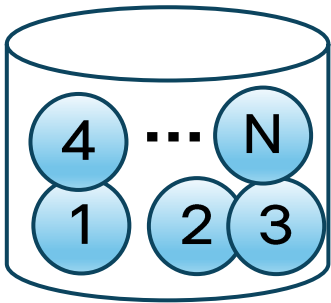


2. 합계 구하기

■ 1부터 자연수 N 까지의 합계를 구하시오.

- 알고리즘?

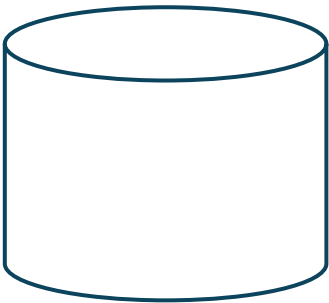
1~N 합계



2. 합계 구하기

- 1부터 자연수 N까지의 합계를 구하시오.
 - 알고리즘 \Rightarrow 반복, 조건문을 활용하여 표현 (의사코드)

1~N 합계 (의사코드)

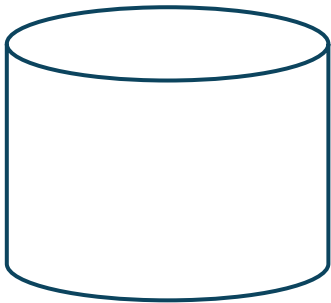


sum = 0

2. 합계 구하기

- 1부터 자연수 N까지의 합계를 구하시오.
 - 알고리즘 \Rightarrow 반복, 조건문을 활용하여 표현 (의사코드)

1~N 합계 (의사코드)



1 2 3 4 ... N

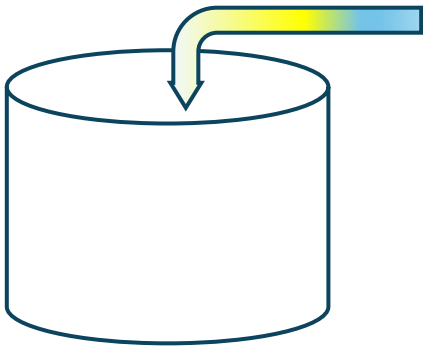
sum = 0

for num in 1-N:

2. 합계 구하기

- 1부터 자연수 N까지의 합계를 구하시오.
 - 알고리즘 \Rightarrow 반복, 조건문을 활용하여 표현 (의사코드)

1~N 합계 (의사코드)



1 2 3 4 ... N

sum = 0

for num in 1-N:

sum = sum + num

2. 합계 구하기

- 1부터 자연수 N까지의 합계를 구하시오.
 - 알고리즘 \Rightarrow 반복, 조건문을 활용하여 표현 (의사코드)

1~N 합계 (의사코드)

```
sum = 0
for num in 1-N:
    sum = sum + num
```

최악 시간복잡도?

2. 합계 구하기

- 1부터 자연수 N까지의 합계를 구하시오.
 - 알고리즘 \Rightarrow 반복, 조건문을 활용하여 표현 (의사코드)

1~N 합계 (의사코드)

```
sum = 0
for num in 1-N:
    sum = sum + num
```

최악 시간복잡도 : $O(N)$

2. 합계 구하기

■ 1부터 자연수 N까지의 합계를 구하시오.

- 알고리즘1 (파이썬)

1~N 합계 알고리즘1 (파이썬)

```
sum = 0
for num in range(1, N + 1):
    sum = sum + num
print(sum)
```

2. 합계 구하기

■ 1부터 자연수 N까지의 합계를 구하시오.

- 알고리즘1 실행 결과

1~10 합계 알고리즘1 (파이썬)

```
N = 10
sum = 0
for num in range(1, N+1):
    sum = sum + num
print(sum)
```

```
In [14]: runfile('D:/data/lecture/
파이썬으로배우는자료구조와알고리즘/code/알고리즘/
untitled1.py', wdir='D:/data/lecture/
파이썬으로배우는자료구조와알고리즘/code/알고리즘')
55
```

2. 합계 구하기

■ 1부터 자연수 N까지의 합계를 구하시오.

- 알고리즘2 (파이썬)

1~10 합계 알고리즘2 (파이썬)

```
def cal_sum(N):
```

```
    sum = 0
```

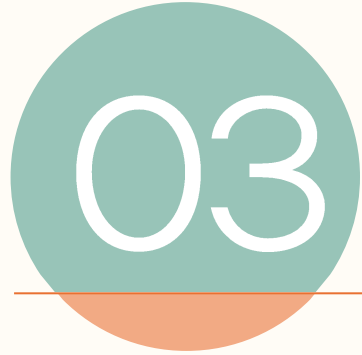
```
    for num in range(1, N+1):
```

```
        sum = sum + num
```

```
    return sum
```

```
print(cal_sum(10))
```

```
In [15]: runfile('D:/data/lecture/  
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘/  
untitled1.py', wdir='D:/data/lecture/  
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘')  
55
```



최대값 찾기

3. 최대값 찾기

■ N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

- 파이썬 리스트

- `list = [5, 3, 2, 1, 7, 9, 6]`

최대값 찾기



3. 최대값 찾기

- N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

의사코드



max = ?

3. 최대값 찾기

- N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

의사코드



```
max = list[0]
```


3. 최대값 찾기

- N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

의사코드



```
max = list[0]
```

```
for num in list[1] ~ list[N-1]:
```

3. 최대값 찾기

- N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

의사코드



```
max = list[0]
for num in list[1] ~ list[N-1]:
    if (max < num):
        max = num
return max
```

3. 최대값 찾기

- N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

의사코드



```
max = list[0]
for num in list[1] ~ list[N-1]:
    if (max < num):
        max = num
return max
```

최악 시간복잡도?

3. 최대값 찾기

- N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

의사코드



```
max = list[0]
for num in list[1] ~ list[N-1]:
    if (max < num):
        max = num
return max
```

최악 시간복잡도 : $O(N)$

3. 최대값 찾기

- N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

파이썬 코드

```
def find_max(list):  
    max = list[0]  
    for num in list:  
        if (max < num):  
            max = num  
    return max  
  
print(find_max([5, 3, 2, 1, 7, 9, 6]))
```

3. 최대값 찾기

- N개의 숫자를 저장한 파이썬 리스트에서 가장 큰 값을 출력하시오.

파이썬 코드 실행

```
def find_max(list):  
    max = list[0]  
    for num in list:  
        if (max < num):  
            max = num  
    return max
```

```
print(find_max([5, 3, 2, 1, 7, 9, 6]))
```

```
In [16]: runfile('D:/data/lecture/  
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘/  
untitled1.py', wdir='D:/data/lecture/  
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘')  
9
```



약수의 개수

4. 약수의 개수

- 자연수 N 의 약수의 개수를 출력하시오.

의사코드

4. 약수의 개수

- 자연수 N 의 약수의 개수를 출력하시오.

의사코드

```
count = 0
```

4. 약수의 개수

- 자연수 N 의 약수의 개수를 출력하시오.

의사코드

```
count = 0
```

```
for num in 1 ~ N:
```

4. 약수의 개수

- 자연수 N 의 약수의 개수를 출력하시오.

의사코드

```
count = 0
for num in 1 ~ N:
    if (N % num == 0):
        count = count + 1
return count
```

4. 약수의 개수

- 자연수 N 의 약수의 개수를 출력하시오.

의사코드

```
count = 0  
for num in 1 ~ N:  
    if (N % num == 0):  
        count = count + 1  
return count
```

최악 시간복잡도?

4. 약수의 개수

- 자연수 N 의 약수의 개수를 출력하시오.

의사코드

```
count = 0  
for num in 1 ~ N:  
    if (N % num == 0):  
        count = count + 1  
return count
```

최악 시간복잡도 : $O(N)$

4. 약수의 개수

- 자연수 N의 약수의 개수를 출력하시오.

파이썬 코드

```
def count_divisors(N) :  
    count = 0  
    for num in range(1, N+1) :  
        if (N % num == 0) :  
            count = count + 1  
    return count  
  
print(count_divisors(10))
```

4. 약수의 개수

- 자연수 N의 약수의 개수를 출력하시오.

파이썬 코드 실행

```
def count_divisors(N):  
    count = 0  
    for num in range(1, N+1):  
        if (N % num == 0):  
            count = count + 1  
    return count  
  
print(count_divisors(10))
```

```
In [18]: runfile('D:/data/lecture/  
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘/  
untitled1.py', wdir='D:/data/lecture/  
파이썬으로 배우는 자료구조와 알고리즘/code/알고리즘')  
4
```

정리하기

- ✓ brute-force 전략 개념
- ✓ 합계 구하는 알고리즘
- ✓ 최대값, 최소값 찾는 알고리즘
- ✓ 약수의 개수 구하는 알고리즘

03강

다음 시간 안내 ▶▶▶

Brute-force 전략2