

Chapter. 02
고급 자료구조

| 핵심 유형 문제풀이

FAST CAMPUS
ONLINE
유형별 문제풀이

강사. 나동빈

Chapter. 02

고급 자료구조(핵심 유형 문제풀이)

I 혼자 힘으로 풀어 보기

문제 제목: SHA-256

문제 난이도: 하(Easy)

문제 유형: 해시, 구현

추천 풀이 시간: 15분(검색 시간 포함)

I 문제 풀이 핵심 아이디어

1. hashlib의 sha256 함수를 이용하면 SHA 256 해시를 구할 수 있습니다.
2. hashlib.sha256(문자열의 바이트 객체).hexdigest() : 해시 결과 문자열

| 소스코드

```
import hashlib

input_data = input()
encoded_data = input_data.encode()
result = hashlib.sha256(encoded_data).hexdigest()
print(result)
```

I 혼자 힘으로 풀어 보기

문제 제목: 수 찾기

문제 난이도: 하(Easy)

문제 유형: 해시, 배열, 구현

추천 풀이 시간: 20분

I 문제 풀이 핵심 아이디어

1. 특정 정수의 등장 여부만을 간단히 체크하면 됩니다.
2. Python에서는 dictionary 자료형을 해시처럼 사용할 수 있습니다.
3. 본 문제는 set 자료형을 이용해 더욱 간단히 풀 수 있습니다.

| 소스코드

```
n = int(input())
array = set(map(int, input().split()))
m = int(input())
x = list(map(int, input().split()))

for i in x:
    if i not in array:
        print('0')
    else:
        print('1')
```


I 혼자 힘으로 풀어 보기

문제 제목: 친구 네트워크

문제 난이도: 중(Medium)

문제 유형: 해시, 집합, 그래프

추천 풀이 시간: 50분

I 문제 풀이 핵심 아이디어

1. 해시를 활용한 Union-Find 알고리즘을 이용해 문제를 풀 수 있습니다.
2. Python에서는 dictionary 자료형을 해시처럼 사용할 수 있습니다.

I 문제 풀이 핵심 아이디어

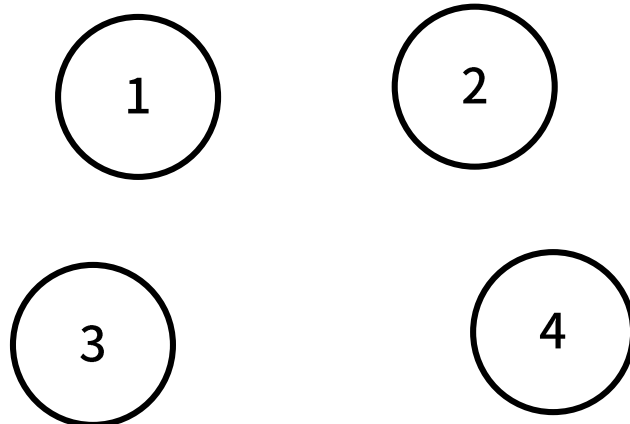
합 집합 찾기(Union-Find) 알고리즘

- 원소들의 연결 여부를 확인하는 알고리즘입니다.
- (그림 가정) 더 작은 원소를 부모로 삼도록 설정

부모 테이블	1	2	3	4
	1	2	3	4

1 - 4

2 - 4



I 문제 풀이 핵심 아이디어

합 집합 찾기(Union-Find) 알고리즘

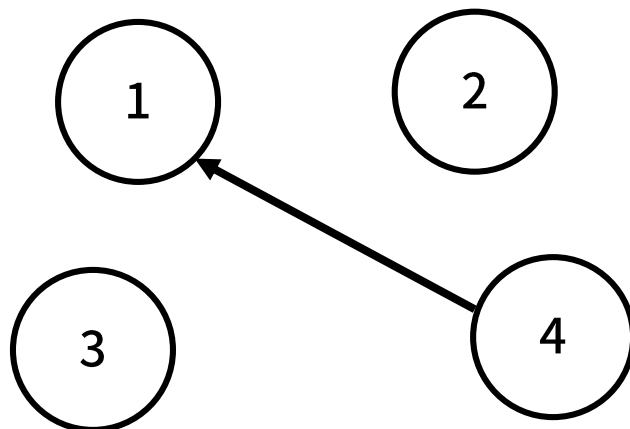
- 원소들의 연결 여부를 확인하는 알고리즘입니다.
- (그림 가정) 더 작은 원소를 부모로 삼도록 설정

부모 테이블

1	2	3	4
1	2	3	1

1 - 4

2 - 4



I 문제 풀이 핵심 아이디어

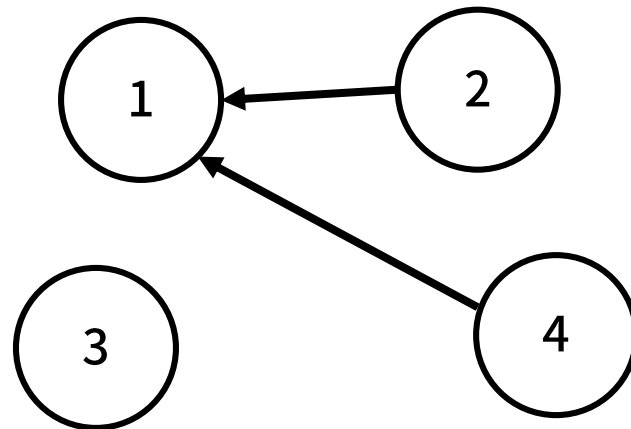
합 집합 찾기(Union-Find) 알고리즘

- 원소들의 연결 여부를 확인하는 알고리즘입니다.
- (그림 가정) 더 작은 원소를 부모로 삼도록 설정

부모 테이블

1	2	3	4
1	1	3	1

1 - 4
(재귀적) 2 - 4



I 문제 풀이 핵심 아이디어

합 집합 찾기(Union-Find) 알고리즘

- 원소들의 연결 여부를 확인하는 알고리즘입니다.
- (그림 가정) 더 작은 원소를 부모로 삼도록 설정

부모 테이블	1	2	3	4
	1	1	3	1

결과: {1, 2, 4}, {3}

I 문제 풀이 핵심 아이디어

합 집합 찾기(Union-Find) 알고리즘

```
def find(x):  
    if x == parent[x]:  
        return x  
    else:  
        p = find(parent[x])  
        parent[x] = p  
        return parent[x]
```

```
def union(x, y):  
    x = find(x)  
    y = find(y)  
  
    parent[y] = x
```

```
parent = []
```

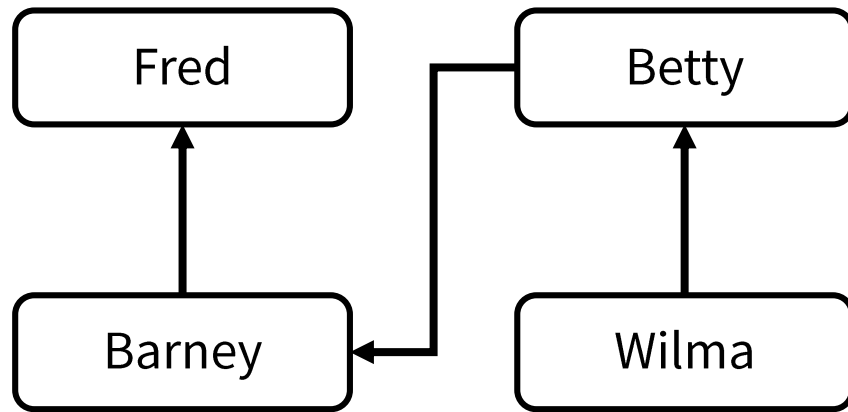
```
for i in range(0, 5):  
    parent.append(i)
```

```
union(1, 4)  
union(2, 4)
```

```
for i in range(1, len(parent)):  
    print(find(i), end=' ')
```

I 문제 풀이 핵심 아이디어

- 1) Fred Barney ✓
- 2) Betty Wilma ✓
- 3) Barney Betty ✓



출력: 2 출력: 2 출력: 4

[부모 테이블]

Fred	Fred
Barney	Fred
Betty	Fred
Wilma	Fred

[네트워크 크기]

Fred	4
Barney	4
Betty	4
Wilma	4

| 소스코드

```
def find(x):
    if x == parent[x]:
        return x
    else:
        p = find(parent[x])
        parent[x] = p
        return parent[x]

def union(x, y):
    x = find(x)
    y = find(y)

    if x != y:
        parent[y] = x
        number[x] += number[y]
```

```
test_case = int(input())

for _ in range(test_case):
    parent = dict()
    number = dict()

    f = int(input())

    for _ in range(f):
        x, y = input().split(' ')

        if x not in parent:
            parent[x] = x
            number[x] = 1
        if y not in parent:
            parent[y] = y
            number[y] = 1

        union(x, y)

    print(number[find(x)])
```