

2021-2 자료구조및실습 실습과제 06

학번	2020136129	이름	최수연
----	------------	----	-----

1) 각 문제에 대한 분석 및 해결 방법

1. 7.3절의 합집합 코드를 참조하여 정렬된 리스트를 이용한 집합에서의 효율적인 교집합 함수를 구현하라.

[문제분석 및 해결방법]

교집합의 경우 새로운 집합을 만들고 이중 while 반복문을 사용하여 집합 A와 집합 B를 하나씩 비교해 만약 같으면 새로운 집합에 그 값을 추가하도록 한다. 해당 while문을 모두 빠져나오면 새로운 집합에는 집합 A와 B의 교집합의 원소가 모이게 된다. 자세한 코드는 2) 자신이 구현한 주요 코드를 참고한다.

2. 차집합도 같은 방법으로 개선할 수 있다. 7.3절의 합집합 코드를 참조하여 정렬된 리스트를 이용한 집합에서의 효율적인 차집합 함수를 구현하라.

[문제분석 및 해결방법]

차집합의 경우 새로운 집합을 만들고 임시로 값을 저장할 idx라는 리스트를 만든다. 그리고 이중 while 반복문을 사용하여 집합 A와 집합 B를 하나씩 비교하면서 만약 같을 경우, 그 값을 idx에 추가하고 내부의 while 문을 빠져나온다. 해당 이중 while 반복문을 모두 빠져나오면, 이중 for 문을 사용하여 idx 리스트와 집합 A를 비교하여 만약 비교하는 값이 같지 않을 경우, 새로운 집합에 추가한다. 다시 말해, 집합 A와 B가 겹치는 값을 idx에 넣고, 다시 집합 A와 리스트 idx를 비교하여 서로 값이 같지 않을 때, 이때 A의 원소를 새로운 집합으로 추가하도록 한다. 이렇게 하면 '집합 A'에서 '집합 A와 B가 겹치는 값'을 뺀 집합이 새로운 집합, 차집합이 된다.

2) 자신이 구현한 주요 코드

```
def intersect(self, setB): # 교집합
    newSet = Set()
    a = 0
    while a < len(self.items):
        b = 0
        while b < len(setB.items):
            valueA = self.items[a]
            valueB = setB.items[b]
            if valueA == valueB:
                newSet.items.append(valueA)
            b += 1
        a += 1
    return newSet

def difference(self, setB): # 차집합
    newSet = Set()
    a = 0
    idx = []
```

```

while a < len(self.items):
    b = 0
    while b < len(setB.items):
        valueA = self.items[a]
        valueB = setB.items[b]
        if valueA == valueB:
            idx.append(valueA)
            break
        b += 1
    a += 1
for i in range(len(idx)):
    for j in range(len(self.items)):
        if self.items[j] != idx[i]:
            newSet.items.append(self.items[j])
            break
return newSet

```

3) 다양한 입력에 대한 테스트 결과

```

Set A: ['손수건', '지갑', '휴대폰']
Set B: ['빗', '야구공', '지갑', '파이썬 자료구조']
Set A: ['지갑', '휴대폰']
Set B: ['빗', '야구공', '지갑', '파이썬 자료구조']
A U B: ['빗', '야구공', '지갑', '파이썬 자료구조', '휴대폰']
A ^ B: ['지갑']
A - B: ['휴대폰']

```

4) 코드에 대한 설명 및 해당 문제에 대한 고찰

코드 설명은 1) 각 문제에 대한 분석 및 해결 방법 참고.

5) 이번 과제에 대한 느낀점

이번 과제는 정렬된 리스트를 사용한 교집합, 차집합을 구현하는 것이었으므로 생각보다 쉬운 것 같았다. 중간고사 준비를 해야 해서 선형조사법을 이용한 해시 맵 선택2 과제는 해보지 못했지만 다른 과목 중간고사가 끝나고 여유로울 때 선택2 과제를 한번 해봐야겠다.

6) 궁금한 점이나 건의사항

딱히 없습니다.

7) 자신이 구현한 전체 코드

```
# Set.
class Set:
    def __init__(self):
        self.items = []
    def size(self):
        return len(self.items)
    def contains(self, item):      #O(n)
        return item in self.items
    def insert(self, elem):        #O(n) --> 수정 --> 여전히 O(n)
        if elem in self.items: return      # 이미 있음
        for idx in range(len(self.items)): # loop: n번
            if elem < self.items[idx]:      # idx위치에 삽입
                self.items.insert(idx, elem)
            return
        self.items.append(elem)            # 맨 뒤에 삽입
    def delete(self, elem):         #O(n)
        self.items.remove(elem)
    def __eq__(self, setB):         #O(mn) --> O(m+n)
        if self.size() != setB.size():
            return False
        for idx in range(len(self.items)): # loop: n번
            if self.items[idx] != setB.items[idx]:
                return False
        return True

    def union(self, setB): # 합집합
        newSet = Set()
        a = 0
        b = 0
        while a < len(self.items) and b < len(setB.items):
            valueA = self.items[a]
            valueB = setB.items[b]
            if valueA < valueB:
                newSet.items.append(valueA)
                a += 1
            elif valueA > valueB:
                newSet.items.append(valueB)
                b += 1
            else: # Only one of the two duplicates are appended.
                newSet.items.append(valueA)
                a += 1
                b += 1
        while a < len(self.items):
```

```

        newSet.items.append(self.items[a])
        a += 1
    while b < len(setB.items):
        newSet.items.append(setB.items[b])
        b += 1
    return newSet

```

```

def intersect(self, setB): # 교집합
    newSet = Set()
    a = 0
    while a < len(self.items):
        b = 0
        while b < len(setB.items):
            valueA = self.items[a]
            valueB = setB.items[b]
            if valueA == valueB:
                newSet.items.append(valueA)
            b += 1
        a += 1
    return newSet

```

```

def difference(self, setB): # 차집합
    newSet = Set()
    a = 0
    idx = []
    while a < len(self.items):
        b = 0
        while b < len(setB.items):
            valueA = self.items[a]
            valueB = setB.items[b]
            if valueA == valueB:
                idx.append(valueA)
                break
            b += 1
        a += 1
    for i in range(len(idx)):
        for j in range(len(self.items)):
            if self.items[j] != idx[i]:
                newSet.items.append(self.items[j])
                break
    return newSet

```

```

def display(self, msg):
    print(msg, self.items)

```

```
setA = Set()
setA.insert('휴대폰')
setA.insert('지갑')
setA.insert('손수건')
setA.display('Set A: ')
```

```
setB = Set()
setB.insert('빗')
setB.insert('파이썬 자료구조')
setB.insert('야구공')
setB.insert('지갑')
setB.display('Set B: ')
```

```
setB.insert('빗')
setA.delete('손수건')
setA.display('Set A: ')
setB.display('Set B: ')
```

```
setA.union(setB).display('A U B: ')
setA.intersect(setB).display('A ^ B: ')
setA.difference(setB).display('A - B: ')
```