

## Homework 1: MapReduce (Chapter 2)

2016-19762 Eunman Park

### Top-50 Follower counting problem

dst\_count.py

```
def map(self, context):  
    dst = context.value.split('\t')[1]  
    context.emit(dst, 1)
```

map(): (src, dst) -> (dst, 1)

dst 를 key 로, 1 을 value 로 하여 내보냈다.

```
def reduce(self, context):  
    cnt = sum(context.values)  
    context.emit(context.key, cnt)
```

reduce(): (dst, [1...]) -> (dst, cnt)

values 의 요소(= 1)를 모두 더해 해당 dst 의 개수를 계산했다.

filter.py

```
def map(self, context):  
    dst, cnt = context.key, context.value  
    cnt = struct.unpack(">i", cnt)[0]  
    context.emit(0, (dst, cnt))
```

map(): (dst, cnt) -> (0, (dst, cnt))

0 를 key 로, (dst, cnt)를 value 로 하여 하나의 reducer 로 내보냈다.

```
def reduce(self, context):  
    dic = {}  
    lst = []  
    for pair in list(context.values):  
        dst, cnt = pair[0], pair[1]  
        dic[dst] = cnt  
    lst = sorted(dic.items(), key=lambda t:t[1],  
reverse=True);  
    for i in range(50):  
        context.emit(lst[i][0], lst[i][1])
```

reduce(): (0, [(dst, cnt)...]) -> 선별된 (dst, cnt)

딕셔너리를 이용하여 cnt 를 기준으로 내림차순으로 정렬한 뒤, 50 개를 출력했다.

## Non-symmetric fellow-ship problem

nsf.py

```
def map(self, context):  
    src = context.value.split('\t')[0]  
    dst = context.value.split('\t')[1]  
    if int(src) > int(dst):  
        src, dst = dst, src  
    context.emit((src, dst), 1)
```

map(): (src, dst) -> ((smaller, bigger), 1)

src 와 dst 의 대소를 판단하여 (작은 것, 큰 것)을 key 로, 1 을 value 로 하여  
내보냈다.

```
def reduce(self, context):  
    cnt = sum(context.values)  
    if cnt == 1:  
        context.emit(context.key[0], context.key[1])
```

reduce(): ((smaller, bigger)...), 1 -> 선별된 (smaller, bigger)

values 의 요소(= 1)를 모두 더해 해당 쌍의 개수를 계산했다. 그리고 개수가 1 인  
쌍, 즉 non-symmetric 한 쌍만을 출력했다.