

Week 23

## 3078. 좋은친구

- 학생들의 이름이 성적순으로 주어졌을 때, 앞뒤로 k 이내에 이름의 길이가 같은 학생 쌍이 몇인지 구하는 문제.
- Queue를 이용. k범위 내에 이름의 길이가 L인 학생수를 저장.
  1. 입력을 queue에 string의 길이로 저장한다.
  2. 이름의 길이가 최대 20글자가 될 수 있으니, length[21] 배열을 만들어서 idx길이를 갖고 있는 학생 수가 몇명인지 저장해둔다.
  3. queue에서 맨 앞 한명씩 빼서 다음 학생을 넣고, length[그 학생의 이름 길이] += 1을 해주고, length[뺀 학생이름길이]를 ans에 더해준다.

## 3078. 좋은친구

3. queue에서 맨 앞 한명씩 빼서 다음 학생을 넣고, length[그 학생의 이름 길이] += 1을 해주고, length[뺀 학생이름길이]를 ans에 더해준다.

```
while dq:
    a = dq.popleft()
    length[a] -= 1
    if len(dq) >= k:
        length[dq[k-1]] += 1
    ans += length[a]
```

# 2287. 모노디지털 표현

- Set을 사용.
- 1자리  $\rightarrow k$
- 2자리  $\rightarrow k+k, k-k, k*k, k/k, kk$
- 3자리  $\rightarrow$  1자리 (+-\*/ ) 2자리 or 2자리 (+-\*/ ) 1자리
- ...
- 이런식으로 보면 다음과 같은 규칙이 나옴.

• # 자리수 만드는데 사용되는 자리수

# 1	1
# 2	11
# 3	21
# 4	31 22
# 5	41 23
# 6	51 42 33
# 7	61 52 43
# 8	71 62 53 44

## 2287. 모노디지털 표현

- Set에서 하나씩 꺼내서 사칙연산을 해주는데, -랑 //은 순서가 결과에 영향을 주기 때문에 두 수를 바꿔서도 해줘야 함.

```
nums = [set() for _ in range(9)]
nums[1].add(k)
tmp = k
for i in range(2, 9):
    for j in range(1, (i//2)+1):
        for a in nums[j]:
            for b in nums[i-j]:
                nums[i].add(a + b)
                if a-b > 0:
                    nums[i].add(a - b)
                if b-a > 0:
                    nums[i].add(b - a)
                nums[i].add(a * b)
                nums[i].add(a // b)
                nums[i].add(b // a)
            if 0 in nums[i]:
                nums[i].remove(0)
    tmp = tmp*10 + k
    nums[i].add(tmp)
```