

Replacement Selection 구현

파일처리론 01분반
컴퓨터공학과 2142978 채종화

목차

1. 구현한 프로그램의 설명
2. 실행 결과
3. 구현에 있어서 어려웠던 점과 해결 방법

프로그램 설명

1. 프로그램의 개요: 이 프로그램은 replacement_input.txt에서 입력받은 값들을 대체 선택으로 정렬하여 터미널과 replacement_output.txt에 출력함.

2. 코드 구성

2.1 replacement_selection 함수

- 버퍼에 숫자를 추가해 정렬
- 버퍼가 가득 차면 가장 작은 값을 추가
- 포함될 수 없는 값은 동결값으로 추가
- 동결 값이 버퍼 크기만큼 쌓이면 새 run 시작
- 남은 값 처리 후 마지막 run 반환

2.2 input 함수

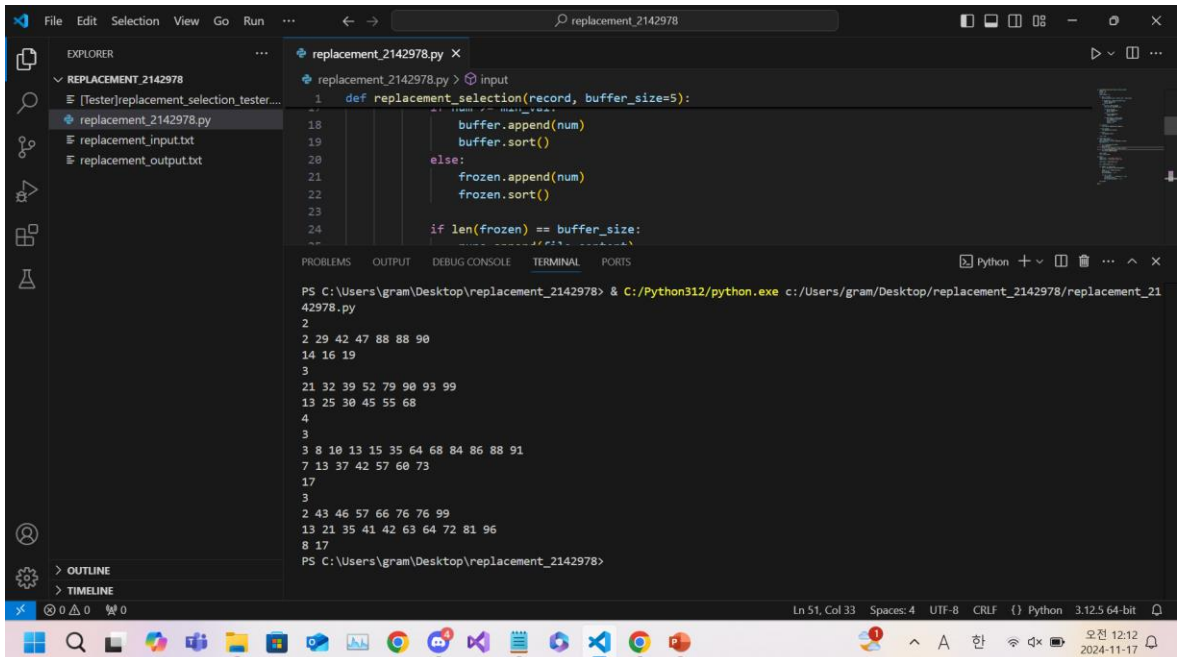
- 입력 파일에서 test_case 개수를 읽음
- 한줄씩 정렬할 값의 개수, 정렬할 정수들 읽음(반복)

2.3 main 함수

- output file에 정렬 결과 출력, 터미널에도 출력

3. 사용한 python version: 3.12.5

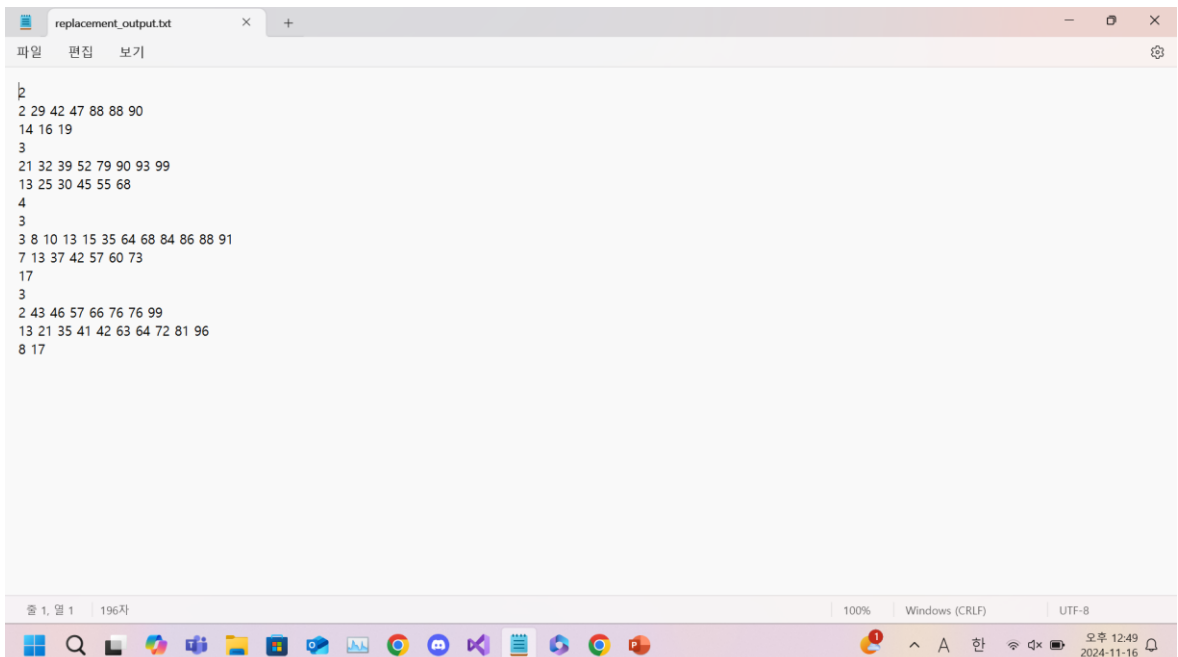
실행 결과



```
def replacement_selection(record, buffer_size=5):
    18     buffer.append(num)
    19     buffer.sort()
    20     else:
    21         frozen.append(num)
    22         frozen.sort()
    23
    24     if len(frozen) == buffer_size:
```

```
PS C:\Users\gram\Desktop\replacement_2142978> & C:/Python312/python.exe c:/Users/gram/Desktop/replacement_2142978/replacement_2142978.py
2
2 29 42 47 88 88 90
14 16 19
3
21 32 39 52 79 90 93 99
13 25 30 45 55 68
4
3
3 8 10 13 15 35 64 68 84 86 88 91
7 13 37 42 57 60 73
17
3
2 43 46 57 66 76 76 99
13 21 35 41 42 63 64 72 81 96
8 17
PS C:\Users\gram\Desktop\replacement_2142978>
```

터미널 결과(vscord)



```
2
2 29 42 47 88 88 90
14 16 19
3
21 32 39 52 79 90 93 99
13 25 30 45 55 68
4
3
3 8 10 13 15 35 64 68 84 86 88 91
7 13 37 42 57 60 73
17
3
2 43 46 57 66 76 76 99
13 21 35 41 42 63 64 72 81 96
8 17
```

생성된 replacement_output.txt

구현에 있어서 어려웠던 점과 해결 방법

-어려웠던 점 1: Buffer에서 동결값이 추가되는 것을 구현하는 것에 어려움을 겪음

-해결 방법: 동결값(frozen)이 추가될수록 buffer내의 effective_buffer가 하나씩 줄어들고, 동결값의 크기가 5가 되면 새로운 run으로 넘어가도록 구현

-어려웠던 점 2: 입력 파일에서 각각의 정보를 어떻게 따로 입력받을지에 대해 어려움을 겪음

-해결 방법: 입력 파일에서 첫 줄을 따로 입력받고, 나머지 줄들은 반복문을 사용함

이상입니다