**파일처리론 리포트**

**replacement selection sorting**

**이름: 윤일준**

**학과: 컴퓨터공학과**

**학번: 2343457**

**구현한 프로그램의 실행방법**

**파이썬 버전 :** **3.11.4**

**변수**

**num은 정렬 횟수**

**n은 정렬할 값의 개수**

**all\_List에는 정렬할 값들이 공백 없이 리스트로 들어간다.**

**run에는 정렬 할 값을 넣어주며 runs에는 지금까지의 모든 run을 저장한다.**

**freeze\_List는 freeze된 값들을 넣어주는 리스트이다.**

**list1은 버퍼 같은 개념으로 all\_List에서 5개의 값을 가져와 최소값을 찾는다.**

**메커니즘**

**Input 파일에서 각각 정렬횟수 정렬 값의 개수 정렬할 값을 받아 변수**

**num,n,all\_List에 삽입한다.**

**list1에 all\_List의 값 5개를 가져온 후 가져온 값을 all\_List에서 삭제한다.**

**list1의 값 중 최소값(min\_element)을 찾아 run리스트에 삽입한다.**

**list1의 최소값 인덱스(min\_index)에 all\_List의 값을 삽입한다.**

**가져온 새로운 값과 run에 들어간 값을 비교하여**

**1.run 값이 큰경우**

**새로운 값을 포함하여 list1의 최소값을 찾아**

**list1이 다 빌때까지 위와 같이 반복한다.**

**2.새로운 값이 큰 경우**

**새로운 값을 freeze\_List에 넣고 list1은 채우지 않는다.**

**list1이 다 빌때까지 위와 같이 반복한다.**

**list1이 다 비었을 경우(값들이 모두 freeze 된 경우)**

**현재 run에 있는 값을 runs에 저장하고**

**freeze\_List에서 버퍼 크기 만큼 정렬을 위해 list1에 채운다.**

**all\_list가 다 비었을 경우**

**freeze\_List의 값을 list1에 채우며 위와 같이 반복한다**

**all\_List와 list1이 비어 run에 숫자가 남아있을 경우**

**runs에 남은 run을 저장한다**

**output.txt 에 출력 방법**

**for r in runs:**

**f.write(" ".join(map(str, r)) + "\n")**

**이 코드로 runs에 저장된 run 리스트의 갯수 만큼 반복하여 각 한줄씩**

**문자열로 출력한다.**

**구현한 프로그램의 실행 결과 스크린샷**

**프로그램 실행 화면**

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**Input**

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**Output**

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

**프로그램 구현중 어려웠던 점과 해결 방법**

**1.freeze된 값을 어떻게 구현할것인가?**

**처음에 코드를 구상하며 freeze 값을 어떻게 구현할 수 있을까 많은 생각을 했었다. 그 중 생각했던 방법이**

1. **freeze된 값에 숫자 1000을 더하여 min 함수에 나오지 않게 하여 모든 원소가 1000이상이 되면 전부 freeze 됐다고 생각하고 다시 1000을 빼주어 재정렬한다.**
2. **freeze된 리스트로 값들을 따로 분류한다.**

**1번 방법으로 했을 경우 과제의 정수 범위가 0에서 100사이의 값이라 1000을 넘어가지 않아 조건문(1000이상인 값은 freeze)에 안 걸려 괜찮다고 생각했으나 일반 값과 freeze 값이 섞이면 복잡해질꺼 같아서**

**2번 방법으로 구현하기로 하였다**

**1.1 해결방법**

**2번 방법으로 구현하여 freeze된 값을 따로 freeze\_List에 넣어주면**

**원래 값을 건들지 않아도 되고 디버깅할떄 더 정확하게 가능했다.**

**2.all\_List에서 list1으로 값을 옮긴 후 all\_List의 값 처리 방법**

**all\_list에서 list1으로 옮긴 값을 삭제 시킬지 유지할지에 대해 생각했다.**

**처음엔 유지시키면서 list1에서 all\_List로 값을 던져줄때 인덱스 값을 증가시키며 값을 옮겨줄까 생각하며 방향을 잡고 나갔으나**

1. **디버깅의 어려움**
2. **다양한 조건에 따른 어려움**

**처음엔 all\_list에서 버퍼크기인 5개의 값**

**min\_element와 next\_value의 비교에 따라**

**all\_List에서 값을 가져오는 경우와 가져오지 않는 두가지 경우에 따라 인덱스로 구현하자 하니 조건문이 많아져 복잡해 질꺼 같았다.**

**2-1 해결 방법**

**all\_List에서 list1으로 보내준 값을 삭제 시켜서**

**값을 채워야하는 경우 all\_List.pop(0)을 통해 all\_List의 값중 맨 처음 값만 보낼 수 있게끔 하는것이 명료할거 같아 가져가는 값을 삭제 시키기로 하였다.**