과제 6: Heap Sort

**주의: 프로그램 소스를 그대로 복사를 하거나 살짝 고쳐서 제출하는 경우 관련 학생들 모두 그 과제에 대해 0점 처리합니다. 두 번 이상 적발되는 경우 전체 과제 점수가 0점이며 D 이하의 학점이 부여됩니다.

1. 개요

Chap8의 heap sorting 프로그램을 작성한다. 아래의 조건을 반드시 준수하여야 한다.

- 파일 I/O 연산은 system call 또는 C 라이브러리만을 사용한다.
- 제공되는 person.h와 heapsort.c를 이용하여 아래의 기능 (3)을 완성한다.

(1) 'Person' 레코드 파일의 구조
- Heap sort를 위해 주어지는 입력 레코드 파일의 구조는 지난 과제 5에서 사용한 파일 구조와 동일하다.
- Heap sort에 의해 새로 생성되는 정렬된 레코드 파일의 구조도 과제 5와 동일하다.

(2) 입력 레코드 파일
- Heap sort를 하기 위해서는 미리 만들어진 레코드 파일이 필요하다. 이를 위해 지난 과제 5에서 만든 '레코드 삽입' 기능을 수행하여 'n' 개의 레코드를 갖는 레코드 파일을 생성하여 입력 레코드 파일로 사용한다. Heap sort에서 정렬의 기준은 주민번호이기 때문에 레코드 삽입 명령어를 수행할 때 입력되는 필드값들 중 주민번호만 중복이 되지 않게 제공해도 무방하다.
- 과제 6에서는 과제 5에서 만든 '레코드 삭제' 기능을 사용하지 않는다.

(3) Heap sort
- 두 종류의 heap 중 Chap8의 heap을 가정한다 (heap의 루트 노도의 킷값이 가장 작다).
- Heap sort는 insert sort의 한 종류이기 때문에 정렬 프로그램에 레코드 파일이 모든 레코드를 저장할 수 있는 메모리 공간이 필요하다. 이러한 공간을 위해 Chap8에서 제시한 '배열'을 사용한다.
- 정렬 프로그램으로 레코드를 읽어 들일 때는 과제 5와 같이 페이지 단위로 읽고, 읽어 온 페이지에 존재하는 레코드를 하나씩 heap에 추가하여 heap을 만들어 나간다. 읽어 온 페이지에서 모든 레코드를 처리하면 그 다음 페이지를 읽어서 위와 같은 방식으로 처리한다.
- 완성한 heap을 이용하여 '주민번호'를 기준으로 오름차순으로 정렬된 새로운 레코드 파일을 만든다. 레코드를 새로운 파일에 저장할 때도 페이지 단위를 사용한다. 페이지를 하나 할당을 받은 후 여기에 레코드를 채운 후 페이지가 다 차면 파일에 쓰기를 수행한다.

  a.out s <input record file name><output record file name>

  <예시>

a.out s person.dat sortedperson.dat

옵션으로 s를 사용하며 레코드 파일 person.dat를 주민번호를 기준으로 오름차순으로 정렬된 sortedperson.dat를 생성한다. 명령의 수행 후 출력은 없다.

** 주의: 정렬된 레코드 파일을 만들 때 반드시 각자 구현한 heap sort를 사용해야 하며, 프로그래밍 언어나 시스템에서 제공하는 라이브러리를 사용하는 것은 절대 금지합니다.

2. 개발 환경

- OS: Linux 우분투 버전 18.0.4
- 컴파일러: gcc 7.5

** 반드시 이 환경을 준수해야 하며, 이를 따르지 않아서 발생하는 불이익은 본인이 책임져야 함

3. 제출물

- 프로그래밍한 소스파일 heapsort.c를 하위폴더 없이(최상위 위치에) zip파일로 압축하여 myclass.ssu.ac.kr 과제 게시판에 제출한다 (모든 제출 파일들의 파일명은 반드시 소문자로 작성). 제공된 person.h는 제출할 필요가 없음.
- 압축한 파일은 반드시 학번_6.zip (예시 20061084_6.zip)과 같이 작성하며, 여기서 6은 여섯 번째 과제임을 의미함

** 채점 프로그램상 오류가 날 수 있으니 꼭 위 사항을 준수하기 바라며, 이를 따르지 않아서 발행하는 불이익은 본인이 책임져야 함