HW9: Merge sort and heap sort

제출은 gitLab을 통해서 하며 **http://hconnect.hanyang.ac.kr/2017\_CSE2010\_수업번호/2017\_CSE2010\_수업번호\_학번.git 에 HW9라는 폴더를 만들어 진행.**

\* 프로그램 제출간 유의사항

- 소스코드에는 주석이 있어야 함.

- 주어진 구조체와 input.txt를 사용해야 함

**- 숙제의 소스코드 평가는 linux ubuntu 16.04.2 LTS 버전 gcc 5.4.0에서 함.**

\* 보고서 제출간 유의사항

- 작성한 소스 코드가 첨부되어야 하며, 실행결과가 첨부되어야 함.

- 분량은 제한이 없으나 1~2apge로 간략하게 설명.

- 보고서는 hw9\_학번.확장자(doc, docx, pdf)로 제출  
 >> Merge sort, heap sort 두 토픽의 내용이 들어가야 함.

**제출시간: '17.6.7(23:59) 까지**

**\*지연제출**

- 24시간 이내는 해당 과제 50% 감점, 48시간 이내는 75% 감점.

- 지연제출자는 E-mail(casualab@hanyang.ac.kr)과 gitlab에 모두 제출.

- E-mail제목: "hw9\_학번\_자신의 수업 요일(수, 목)\_이름 " 형식으로 제출.

**Topic1: Merge sort**

합병정렬을 *재귀적*으로 구현하며, 정렬된 결과를 출력하면 된다.

1. Input

주어진 입력 파일(input1.txt)에서 입력데이터를 가져오며, 예상출력은 아래와 같다.

Input1.txt

|  |
| --- |
| 26 5 77 1 61 11 59 15 48 19 |

예상출력:

|  |
| --- |
|  |

2. Merge sort ADT

void merge(int list[], int left, int mid, int right);

- 입력: int list[]는 매개변수 int left, int mid, int right 인덱스로 분할 영역이 정해져 들어온 배열.

int left는 list[]의 시작 인덱스

int mid는 분할된 영역을 정하는 기준(left부터 mid까지, mid+1부터 right까지 두 개의 블럭)

int right는 list[]의 마지막 인덱스

- 하는 일: 병합을 하며 정렬을 수행한다. 첫 번째 단계에서는 크기가 1인 list[]를 합병하고 두 번째 단계에서는 크기가 2인 리스트를 합병하고 i번째에서는 2i-1인 리스트를 합병한다. 재귀적으로 호출되기에 list[]는 정렬된 상태로 다시 호출되도록 해야 한다.

- 출력: void

void merge\_sort(int list[], int left, int right);

- 입력: int list[]는 메인함수에서 input1.txt를 읽어온 값이 있는 배열.

int left는 list[]의 시작 인덱스.

int right는 list[]의 마지막 인덱스.

- 하는 일: Left, right로 받은 인덱스를 통해 mid(left+right/2)를 계산하고 2개의 merge\_sort(list[], left, mid), merge\_sort(list[], mid+1, right)를 재귀적으로 호출한다. 이렇게 list의 구간을 줄여나가면서 left >= right로 기저(list의 left ~ right구간이 하나의 값만 가질 경우)에 도달했을 경우 더 이상 줄이지 않고 반환해 2개의 list 구간을 계산 한 후 위의 merge함수에 left, mid, right 변수를 대입해 2개의 list 구간을 sort한다.

- 출력: void

3. 프로그램 설명

* 파일명: HW9\_1\_학번.c
* 입력: 주어진 입력 파일(input1.txt)의 숫자("1. input"항목을 참고)
* 출력 : 일반 입출력

**Topic2: Heap Sort**

힙정렬을 구현하고 정렬된 결과를 출력하면 된다. 최대 힙을 활용하게 된다.

1. Input

주어진 입력 파일(input2.txt)에서 입력데이터를 가져오며, 예상출력은 아래와 같다.

Input2.txt

|  |
| --- |
| 73 21 578 109 410 53 51 1 7 45 |

예상출력:

|  |
| --- |
|  |

2. Heap sort ADT

void adjust(int heap[], int root, int n)

- 입력: heap[]은 최대힙의 성질을 만족하지 않는 배열.

int root은 힙정렬을 시작하는 노드의 인덱스

int n은 heap[]의 요소 갯수.

- 하는 일: 왼쪽 및 오른쪽 서브트리 모두가 히프인 이진트리에서 시작하여 이진트리 전체가 최대 히프가 되도록 heap[]을 재조정한다.

- 출력: void

void heap\_sort(int list[], int n)

- 입력: int list[]는 메인함수에서 input2.txt를 읽어온 값이 있는 배열.

int n은 list[]요소의 갯수.

- 하는 일: adjust함수를 반복적으로 호출하여 list[]를 최대 힙으로 만든 다음 힙의 첫 번째 노드와 마지막 노드를 교환한다. 첫 번째 노드는 최대키를 가지므로 교환은 이 최대키를 가진 노드를 list[] 배열의 정확한 위치로 들어가게 한다. 그 다음, 힙의 크기를 줄인 후 다시 adjust함수를 사용해 힙을 조정한다. 교환, 힙 크기 축소, 히프 조정(adjust) 과정을 n-1번 반복하여 list를 sort한다.

- 출력: void

3. 프로그램 설명

* 파일명: HW9\_2\_학번.c
* 입력: 주어진 입력 파일(input1.txt)의 숫자("1. input"항목을 참고)
* 출력 : 일반 입출력