

## 2021년 1학기 자료구조론 과제 #3

### ■ 과제내용 및 명세

#### ▲ 문제 : Min Heap 구현

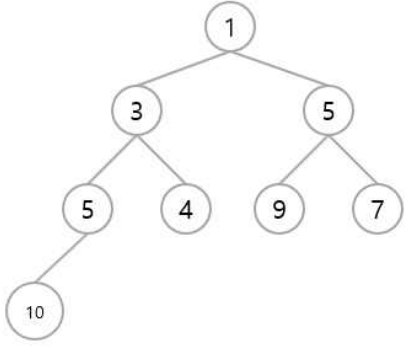
- 교재 Heap 참고
- Heapify, Insert, Delete 기능 구현
- Data : positive integer만 사용
- 파일 입.출력
- 사용 언어 : C
- ※ Visual Studio 권장

#### ▲ 명세(Specification)

- 입출력 : txt 파일로 Min Heap의 elements를 입.출력 한다.
  - 입출력용 파일 명

입력	출력
input.txt	output.txt

- \* 간혹 C:\나 특정 폴더 내에 입력 혹은 출력 파일을 생성하도록 하는 분이 계시는데, 상대 폴더 위치로 지정하셔서 자신의 폴더 내에 해당 파일이 생성되도록 해 주시기 바랍니다. (단 순히 “input.txt”나 “output.txt”로 지정하면 소스 프로그램과 같은 폴더 내에 생성됩니다.)
- 파일 명은 반드시 위에 주어진 이름을 사용하고 불가피할 경우 사유와 사 용한 각 파일에 대한 설명을 문서에 포함하도록 한다.
- 입출력 파일의 형식은 아래의 예와 동일한 형태를 가져야한다.  
 아래의 예와 다르게 할 경우 감점사항에 해당하며 입출력 파일의 형식에 대한 상세한 설명을 문서에 포함해야 한다.
- 입력 형식 : □ 는 data를 의미
  1. 맨 첫 번째 줄은 heapify 되지 않은 initial 데이터가 띄어쓰기로 구분되어 있다.
  2. insertion : i□
  3. deletion : d
- 입력 파일 예 (input.txt)

파일 형식	입력의 최종 결과 (참고)
5 1 7 i3 i4 i9 i5 i10 d d d	 <p>아래 그림 : delete 기능 수행 전의 min heap</p>

※ input 방법을 반드시 document나 source code내에 명시 할 것.

- 각 기능의 동작

1. Min Heapify

- Heapify 되지 않은 initial data를 min heap의 형태로 재 정렬하는 기능이다.
- Recursive Min Heapify (재귀 함수 사용), Sequential Min Heapify (반복문 사용) **두 가지 버전으로 구현해야 한다.**
- **input.txt의 맨 첫 번째 줄의 데이터를 읽은 후 이를 min heapify 함수로 재 정렬한다.** 그 후, 정렬된 min heap에 대해 insertion, delete을 수행한다.

2. Insert (i□)

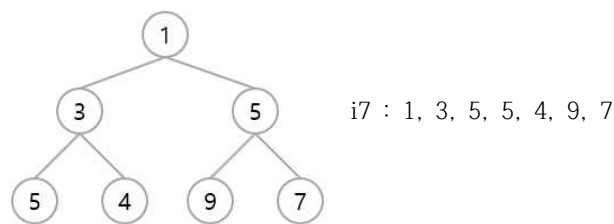
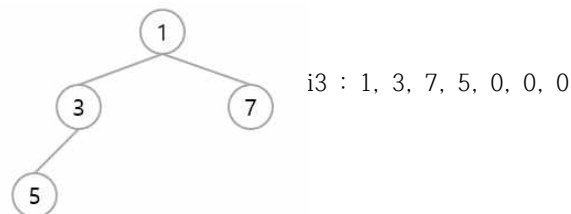
- Min Heap에 □ 삽입
- 데이터 중복 시 Min heap의 성질을 만족하도록 삽입하도록 한다.
- 기본 구현 : Sequential
- **Recursive function으로 구현할 시 추가 점수를 부여한다.**

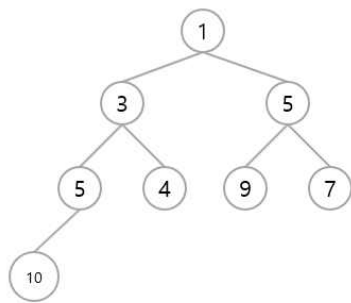
3. Delete (d)

- Min Heap에서 가장 작은 값의 노드(A)를 삭제하고 A의 값을 반환하도록 한다.
- 기본 구현 : Sequential
- **Recursive function으로 구현할 시 추가 점수를 부여한다.**

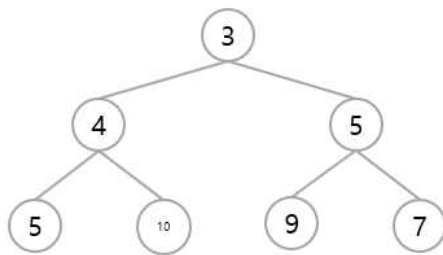
- 출력 형식("해당 입력 command : heap")

1. 입력 파일의 명령과 결과를 **level-order**로 출력하도록 한다.
2. level order로 순회 시 비어있는 노드의 경우 0을 출력 (아래 예시 참고)
3. Delete의 경우 delete 연산의 반환 값인 **트리의 가장 작은 요소 출력 / heap을 level-order로 출력한다.**(자세한 사항은 아래 예시 참고)
4. Delete의 경우 Heap이 비어있는 경우 "Not Exist" 출력





i10 : 1, 3, 5, 5, 4, 9, 7, 10, 0, 0, 0, 0, 0, 0, 0



d : 1 / 3, 4, 5, 5, 10, 9, 7

- 출력 파일 예 (output.txt)

입력 파일의 각 command에 대한 각각의 결과를 단계마다 모두 출력한다.

파일 형식
i3 : 1, 3, 7, 5, 0, 0, 0
i4 : 1, 3, 7, 5, 4, 0, 0
i9 : 1, 3, 7, 5, 4, 9, 0
i5 : 1, 3, 5, 5, 4, 9, 7
i10 : 1, 3, 5, 5, 4, 9, 7, 10, 0, 0, 0, 0, 0, 0, 0
d : 1 / 3, 4, 5, 5, 10, 9, 7
d : 3 / 4, 5, 5, 7, 10, 9, 0
d : 4 / 5, 7, 5, 9, 10, 0, 0

○ 기본구현 :

- Min Heap 구현
- Insert, Delete, Min Heapify 기능 구현

#### ▲ 채점기준

- 각 기능의 정확한 동작
- Documentation

## ■ 과제제출

### ▲ 제출기한 : 2021년 6월 6일 (일)

※ 2021년 6월 6일(일) 자정까지 upload된 과제는 인정합니다.

### ▲ 제출해야 하는 결과물 : source, document, data

1. source(.c 파일) : 상세한 comment를 달 것
2. document (파일이름 : "과제3\_학번\_이름.doc or .hwp")
  - △ 과제 내용 간략하게 요약
  - △ 구현환경 : 구현에 사용한 언어와 compiler, compile 방법 등을 포함
  - △ 설명 : source의 구조와 입.출력 방법 및 Min Heap자료구조와 각각의 function에 대한 상세한 설명
  - △ 자신이 실행해 본 sample 입력과 결과를 document에 첨부
  - △ 기타 : 참조한 source나 구현하지 못한 기능이 있을 경우 언급할 것
3. input : 자신이 구현한 모든 함수들을 테스트 해 볼 수 있는 입력 파일 ("input.txt")
  - 예시의 입력파일과 동일한 형식
4. output : 입력파일에 대한 예상 결과물 ("exp\_output.txt")
  - 프로그램을 실행해서 나온 결과물("output.txt")
  - 예시의 출력파일과 동일한 형식

### ▲ 제출방법 :

1. 자신의 학번과 이름으로 새 폴더 생성 (예:2020004005 홍길동)
2. source code와 document, data(input, output)를 폴더에 넣은 후, 폴더 자체를 압축 (virus check 권장)
3. 블랙보드에 '[과제3]2020004005\_홍길동'으로 과제번호와 자신의 이름을 명시한 후 제출. 잘못 upload하여 다시 올릴 경우나 더 추가하여 올릴 경우에는 '[과제3] 2020004005\_홍길동\_ver2'식으로 version관리를 해 주시기 바랍니다. (최근 version으로만 채점하겠습니다.)

▲ 위의 제출 방법을 지키지 않은 경우 (제출물을 압축하지 않은 경우, 제출 파일의 이름이 다른 경우 등) 감점 사항에 해당합니다.

## ■ 주의사항

▲ 동일한 코드임이 확인될 경우, 같은 코드를 제출한 학생들 모두 0점 처리되니 주의 바랍니다.

▲ 「명세」와 「과제제출」에 명시된 것을 반드시 준수하시기 바랍니다. (지켜지지 않았을 경우 감점)

▲ 알고리즘들의 구현 중 참조한 source가 있을 경우 document에 그 출처와 참조 정도를 반드시 언급해 주시기 바랍니다.

▲ 추가 질문사항은 하단에 명시된 담당조교에게 email로 제출해 주시기 바랍니다.

담당조교 : 공업센터본관 303호 임베디드보안시스템 연구실 정혜민(hmjung2@hanyang.ac.kr)