

## 2019-2 임을규 교수님 알고리즘및문제해결기법

### Programming Assignment #1

과제 제출: **두 개의 소스코드**(.c, .cpp), **두 개의 실행파일**(.exe, elf 등), **보고서**(.docx, .hwp, .pdf 등)를 압축하여 **블랙보드**에 업로드 (ex. 1-1.c, 1-2.c, 1-1.exe, 1-2.exe, report.pdf)

보고서: **1페이지 이상**으로 컴파일 환경과 실행 화면 캡처 등을 작성 (자유 양식)

파일명: **HW1\_학번.zip**으로 할 것

제출 기한: **9월 25일 수요일 14:30**까지 (기한 이후 제출은 0점처리)

인터넷을 참고한 알고리즘 공부는 권장하나, 코드 작성은 본인이 직접 할 것. 소스코드 유사도 검사하여, **베낀 과제는 0점 처리**

문의 사항: 장준영 조교, [lartist@hanyang.ac.kr](mailto:lartist@hanyang.ac.kr) (제출 관련 문의 등)

## 1-1. Heap sort (Introduction to Algorithm 6장)

표준입력(stdin)으로 입력된  $n$ 개의 자연수를 **max heap**으로 정렬하여 내림차순으로 표준출력(stdout)을 통해 출력하라.

- ✓ 각 원소는 0 이상 1000 미만의 값을 가짐
- ✓  $1 \leq n < 100$
- ✓ -1을 입력할 때까지 계속해서 입력 받음

input, output 예시	(stdio)
Input numbers to sort:	
9 45 871 23 13 13 88 46 12 51 99 -1	
Result:	
871 99 88 51 46 45 23 13 13 12 9	

hint

**binary tree:** left, right 최대 두 개의 자식 노드를 가지는 트리

**complete binary tree:** 이진 트리의 원소를 왼쪽 위부터 차례대로 빠짐없이 채운 트리. 빠짐없이 채웠기 때문에 배열에 값을 채우기 쉬우며, 배열의 인덱스를 1번부터 채우는 경우 1보다 큰  $n$ 번째 노드를 기준으로 항상 부모 노드는 인덱스  $\lfloor n/2 \rfloor$ 이며, 자식 노드는 인덱스  $2n$ 과 인덱스  $2n+1$ 이다.

**binary heap:** 완전 이진 트리 중 부모가 자식보다 항상 큰 값을 갖거나(max heap) 혹은 항상 작은 값을 가지는(min heap) 트리. 따라서 루트 노드는 항상 최댓값 또는 최솟값을 가진다.

### heap sort

1. 정렬할 값을 순서 상관없이 완전 이진 트리에 배치한다. (배열 사용)
2. 마지막(가장 오른쪽 아래) 노드부터 이진 힙의 규칙(부모-자식 값 크기 관계)에 맞도록 루트 노드까지 값을 비교하며 재배치한다. 모든 노드가 재배치를 마치면 이진 힙이 완성된다.
3. 최대 또는 최소인 루트 노드를 마지막 노드와 자리를 바꾸고, 이진 힙의 크기를 1 줄인 것으로 간주하여 다시 마지막 노드부터 모든 노드를 재배치한다. 이를 끝까지 반복한다.

## 1-2. 연습문제 6-5.9

위의 과제 1-1 을 참고하여, 오름차순으로 정렬된 배열  $k$  개의 모든 값을 오름차순으로 정렬된 배열 한 개로 합치는,  $O(n \cdot \lg(k))$ 의 시간복잡도를 갖는 알고리즘을 구현하라. ( $n$ : 모든 입력 배열의 원소 개수의 총합)

- ✓ 각 원소는 0 이상 1000 미만의 값을 가짐
- ✓ 각 배열의 원소의 개수는 100개 미만
- ✓  $1 \leq k < 100, 1 \leq n < 10000$

input 예시 (input1-2.txt)	output 예시 (stdout)
4	0 0 1 2 3 5 5 5 6 7 7 10 10 11 13 15 17 18 21 23 30
0 2 3 5 5 10 30	비교안하고 무작위로 넣고 힙솔트하는것이 더 빠를것 순서대로 받으면 시간복잡도가 달라짐 이차원배열같은걸로 그 자체를 받아서 앞에서 세로세로순으로 넣을것
0 5 6 7 10 13 15	
1 7 11	
17 18 21 23	

- ✓ 파일입출력을 사용
- ✓ 입력파일 input1-2.txt는 실행 파일과 같은 경로상에 있는 것으로 가정
- ✓ input1-2.txt의 첫 줄에는  $k$  값, 나머지  $k$  줄에는 각 배열의 원소 나열
- ✓ 출력은 stdout(표준출력)에 정렬된 순서대로 출력
- ✓ 힙, 정렬 관련 라이브러리 사용 금지

hint

정렬된  $k$ 개의 배열에서 한 개씩 원소를 뽑아 최대  $k$ 개의 개수를 갖는 최소 힙을 만드는 시간복잡도는  $O(k)$ 이며, 만들어진 힙에서 최소 원소를 뽑아내고 힙의 구조를 복원하는 시간복잡도는  $O(\lg(k))$ , 해당 원소가 속해 있던 원래의 정렬된 배열에서 다음 값을 힙에 추가하고 힙을 다시 정렬하는 시간복잡도는  $O(\lg(k))$ 이다. 힙에서 총  $n$ 번 원소를 뽑아내야 연산이 종료되므로, 모든 과정의 시간복잡도는  $O(k + n \cdot 2 \cdot \lg(k)) = O(n \cdot \lg(k))$ 이다.