# 알고리즘 설계과제 1

## [HeapSort 비교]

## 1. 과제 개요

프로그래밍에 자신이 넘치는 길동이는 수업시간에 배운 Accelerated Heapsort와, Heapsort를 직접 구현하여 수행시간을 비교 하고자 한다. 단 이때 사용하는 Heap은 수업시간에 배운 대로 **최대 힙(max heap)**을 사용하며, 제자리(in-place)정렬 방식을 사용한다.

## 2. 입력 파일 정보

- 첫 번째 줄에는 테스트케이스의 수 $T$가 입력된다.
- 각 테스트케이스의 첫 번째 줄에는 정렬할 정수의 개수 $N$과 $N$개의 정수($n_1 n_2 ... n_N$)가 공백 하나를 사이에 두고 차례로 입력된다.
- 각 테스트케이스의 두 번째 줄에는 두 정수 $M_1$과 $M_2$가 입력된다.

## 3. 출력 내용 및 기타정보

### (0) 힙 생성

- 모든 정보를 입력받은 뒤, 수업시간에 배운 **선형시간 알고리즘**을 사용하여 힙을 생성한다. 힙의 인덱스는 1부터 시작한다.

### (1) 각 테스트케이스의 첫 번째 줄

- 출력형식: D K
- 설명
    1) D: HeapSort를 사용하여 $M_1$번째로 큰 수를 힙에서 삭제한 뒤 힙을 재구성할 때, **fixHeap을 마지막**으로 호출한 노드의 depth를 출력한다. fixHeap을 호출하지 않을 경우 -1을 출력한다.

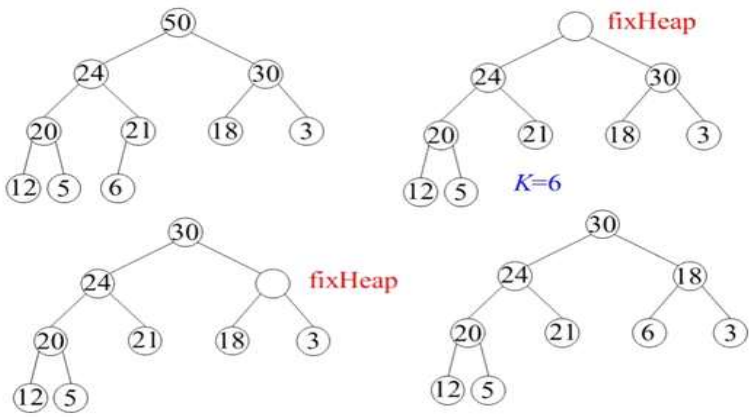


그림 1 . 위 예시는 첫 번째로 큰 수인 50을 힙에서 삭제한 뒤 힙을 재구성하는 과정이다. 이때 fixHeap은 표시된 위치에서 두 번 호출되므로 마지막 fixHeap을 호출한 노드의 depth인 1을 출력한다.

2) K : Accelerated HeapSort를 사용하여, $M_2$번째로 큰 수를 찾을 때 **첫 번째** bubbleUpHeap을 발생시키는 위치에 원래 저장되어 있던 값을 출력한다. bubbleUpHeap이 발생하지 않을 경우, 0을 출력한다. 함수의 진행은 아래 의사코드(pseudocode)와 같다.

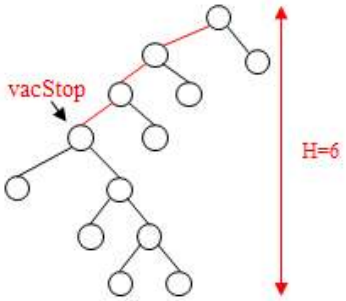


그림 2

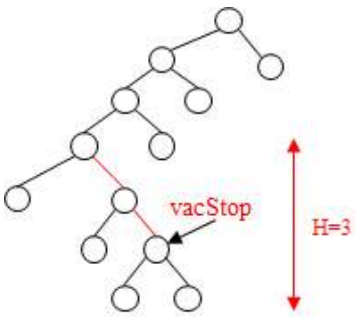


그림 3

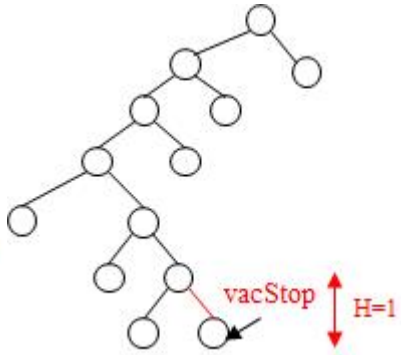


그림 4

```
void fixHeapFast(Element[] E, int n, Element K, int vacant, int h)
{
    int hStop = h/2
    int vacStop = promote (E, hStop, vacant, h);
    int vacParent = vacStop / 2;
    if (E[vacParent].key ≤ K.key)
        E[vacStop] = E[vacParent];
        bubbleUpHeap (E, vacant, K, vacParent);
    else
        fixHeapFast (E, n, K, vacStop, hStop);
}
```

의사코드의 설명은 다음과 같다.

1. hStop 값을 계산한 뒤 promote 함수를 통해서 vacStop 위치를 계산한다.
   이때 hStop 값은 다음과 같은 방법에 의해 정해진다.
   i ) H의 값이 짝수 일 경우, H/2 위치를 vacStop으로 한다.
   ii ) H의 값이 홀수 일 경우, H/2의 값을 올림 한 위치를 vacStop으로 한다.
   ex)
   그림 2의 경우 H의 값이 짝수 이므로, 빨간 간선을 따라서 H/2 위치를 vacStop으로 한다.
   그림 3과 4의 경우 H의 값이 홀수 이므로, H/2의 올림 한 값의 위치를 vacStop으로 한다.

2. vacStop 위치의 노드에서 부모노드의 값과 비교를 한다.
3. 부모노드보다 값이 클 경우, bubbleUpHeap 진행
4. 부모노드보다 값이 작을 경우, 나머지 Heap에 대해서 fixHeapFast 진행

### (2) 각 테스트케이스의 두 번째 줄

- 출력형식: T I
- 설명
    1) T: Accelerated HeapSort의 수행시간을 출력한다.
    2) I: 기존 HeapSort의 수행시간을 출력한다.

*** 프로그램 수행시간 T 계산 및 출력 방법**

```
#include<time.h>

int main(){
        clock_t start_time, end_time;
        double elapsedTime;
        start_time = clock();

        // 측정할 소스코드 부분

        end_time = clock();
        elapsedTime = (double)(end_time - start_time) / CLOCKS_PER_SEC;
        cout << elapsedTime;

        return 0;
}
```

# 4. 입출력 제한사항

(1) 테스트케이스의 $T$의 개수는 최대 1,000개 이다. $(1 \leq T \leq 1{,}000)$
(2) 정수 $N$은 최대 10,000,000개가 입력된다. $(1 \leq N \leq 10{,}000{,}000)$
(3) $M_1$과 $M_2$는 $N$의 범위를 따르게 된다. $(M_1, M_2 \leq N)$
(3) 각 테스트케이스는 1초의 제한시간 이내에 수행되어야한다.
(4) 출력 형식은 표준 입출력을 사용하여 출력한다.

# 5. 프로그램 입출력 예

(1) 표준입출력 예시

빨간색: 프로그램 출력 내용
파란색: 질의 입력 내용

==========

2
100 5 63 18 22 14 35 48 16 82 74 90 86 73 87 41 1 24 38 13 96 33 10 97 23 21 31 99
81 61 30 71 3 52 19 4 66 84 47 11 6 9 89 91 88 50 54 58 65 95 29 17 2 80 62 64 27 68
8 55 34 78 56 37 98 94 76 83 79 85 44 12 36 7 72 93 57 70 40 25 46 100 42 39 26 15
69 51 67 59 43 20 77 28 45 32 53 75 92 49 60

93 41
1 0
0 0 //입력 개수가 적어 0초로 출력됨
100 15 40 42 76 90 47 85 34 49 30 91 22 66 44 41 10 62 16 20 5 9 73 19 86 82 32 50 2 56 83 52 98 71 28 88 78 55 81 12 68 17 25 31 23 7 43 75 29 67 87 33 8 57 36 3 45 100 61 37 79 11 26 65 60 54 96 51 21 39 24 35 46 69 72 13 4 89 77 95 74 99 97 93 92 94 80 58 53 38 64 48 6 84 14 70 59 63 27 1 18
94 11
1 0
0 0 //입력 개수가 적어 0초로 출력됨

# 6. 주의 사항 (지키지 않으면 불이익이 있을수 있습니다)

## (1) 프로그래밍 환경

① OS: Windows, Linux
② 언어: C, C++
③ 개발 환경: Visual Studio 2015 이상(Windows), Eclipse CDT(Linux)

## (2) 제출 파일

다음 파일들을 **"학번 이름 1차.zip"** 형식으로 압축합니다. (예 : 12059876 홍길동 1차) 문서와 프로젝트의 이름도 **"학번 이름 1차"로 통일합니다.**

① 보고서
   (a) 형식: 아래아한글 문서(.hwp), MS Word 문서(.doc, .docx), PDF 문서(.pdf)
   (b) 양식: 첨부된 파일 참조
② 소스 코드가 포함된 프로젝트
   (a) 알고리즘의 핵심 코드처럼 중요한 부분에는 **반드시 주석이 기재돼 있어야 합니다.** 다른 곳에도 최대한 자세히 써 주세요. 또한 하나의 파일(.cpp 혹은 .c)에 코드를 작성해주세요.
   (b) 소스 코드가 포함된 프로젝트 폴더 전체를 압축해서 제출하세요.

## (3) 기타

① 제출 마감은 **5월 12일 금요일 오후 5시 30분 0초**입니다. **마감 후에 제출되는 과제는 받지 않습니다.**
② 마감 직전에는 I-Class에 학생들이 많이 몰리기 때문에 서버가 혼잡할 수 있습니다. 마감 시간보다 1시간 정도 이전에 여유 있게 제출해 주세요.
③ 제출 후에는 제대로 제출되었는지 반드시 확인하시기 바랍니다.
④ 부정행위가 적발될 경우에는 베낀 학생은 물론 **원본을 제공한 학생도 0점 처리합니다.** 인터넷에 올라와 있는 코드나 프로그래밍 참고서의 예제, 예전에 제출됐던 과제 등을 베껴서 제출해도 **부정행위로 처리될 수 있습니다.**