

3주차 : Priority Queue

알 고 리 즈

2018. 9. 21.

멀티미디어 SoC 및 시스템 연구실
민기홍

Overview

▶ Heap (힙)

- Heap이란? - 사전적 의미 : 더미 덩어리 모래산
- 최대 힙(Max Heap)과 최소 힙(Min Heap)
- 힙의 구현

▶ Priority Queue (우선 순위 큐)

- 우선 순위 큐란?
- 우선 순위 큐의 주요 기능 및 구현 원리

▶ 실습 / 과제

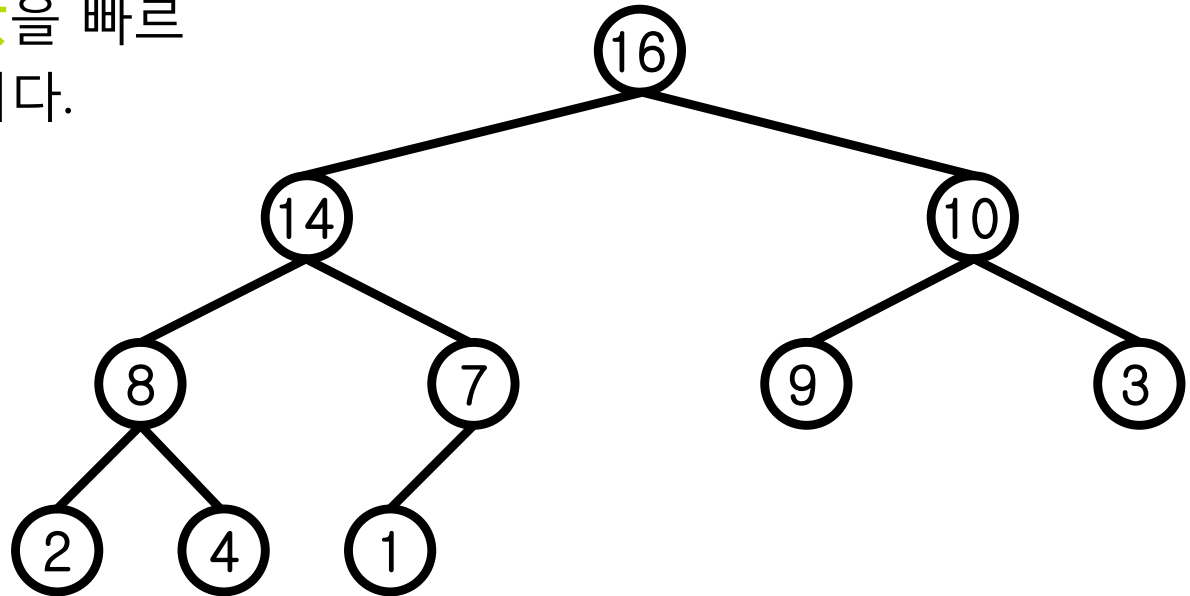
우선 순위와 **작업명**을 갖는 Priority Queue 구현

Heap

▶ Heap이란? (1/3)

완전 이진 트리를 기반으로 하는 배열형 자료구조.

최대값 또는 최소값을 빠르게 찾는데 효율적이다.

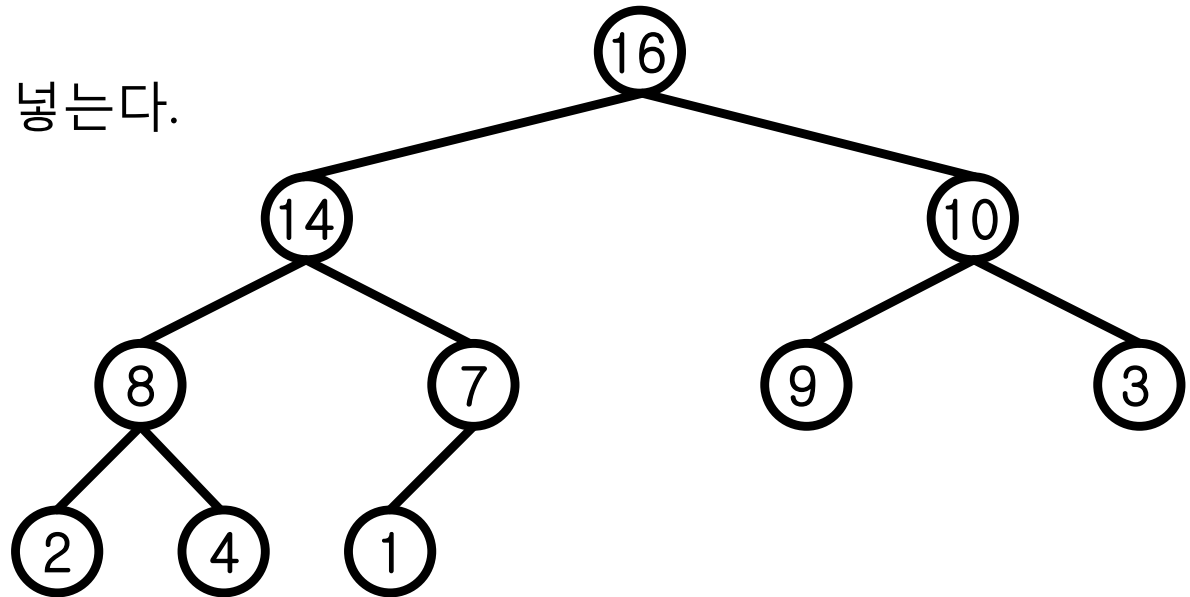


16	14	10	8	7	9	3	2	4	1
1	2	3	4	5	6	7	8	9	10

Heap

▶ Heap이란? (2/3)

이 트리는 가장 낮은 층을 제외하고는 완전히 차 있고,
가장 낮은 층은
왼쪽에서부터 채워 넣는다.



16	14	10	8	7	9	3	2	4	1
1	2	3	4	5	6	7	8	9	10

Heap

▶ Heap이란? (3/3)

어떤 노드가 저장된 인덱스를 i 라 할 때,

PARENT(i)

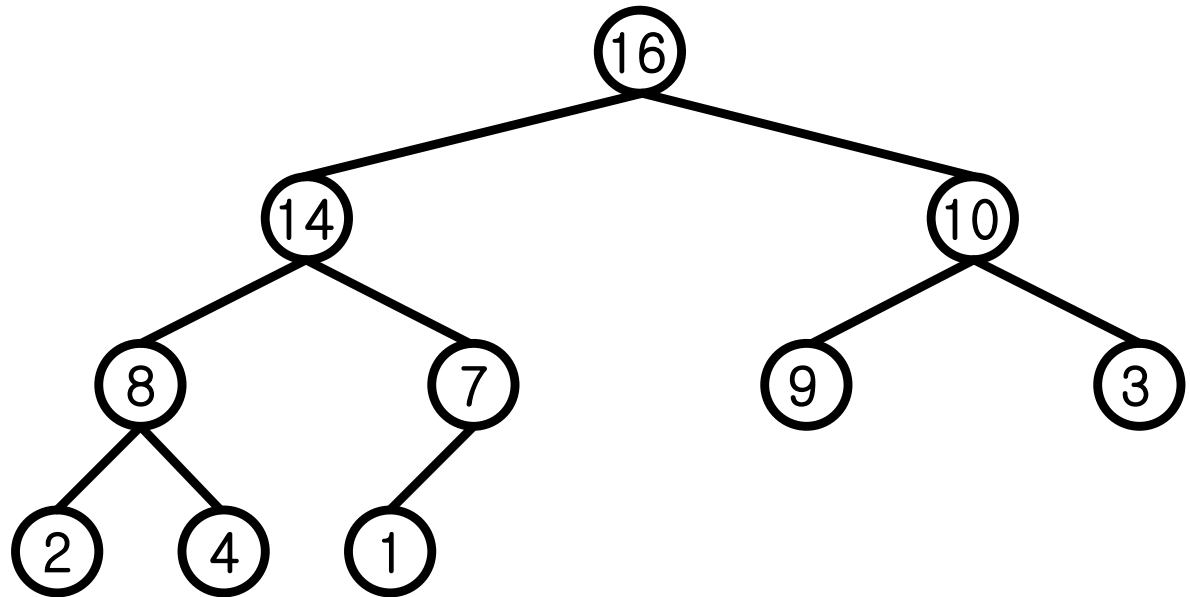
return $\lfloor Li/2 \rfloor$

LEFT-CHILD(i)

return $2i$

RIGHT-CHILD(i)

return $2i+1$



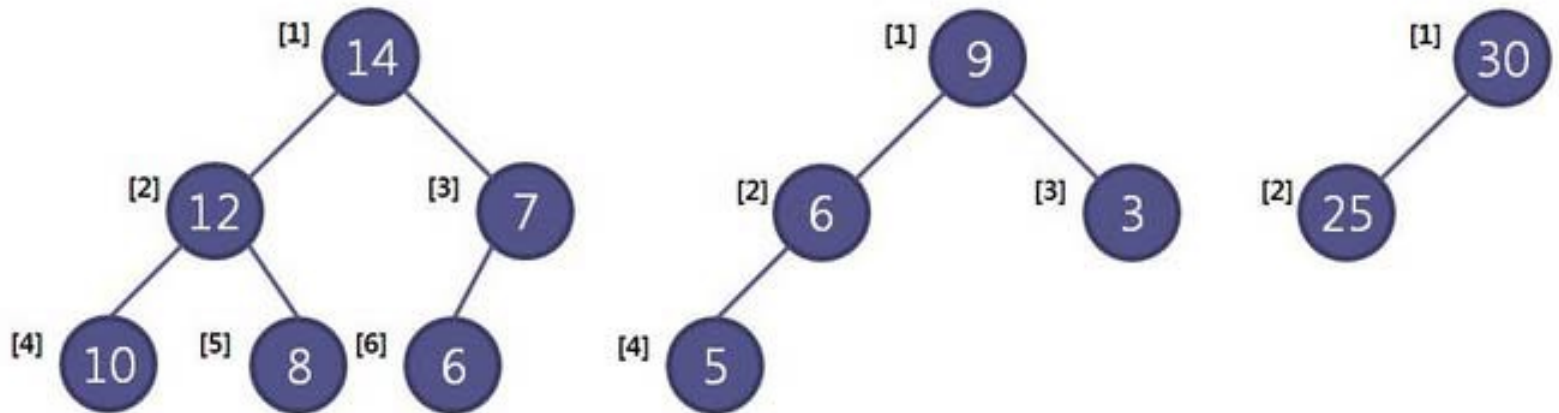
16	14	10	8	7	9	3	2	4	1
1	2	3	4	5	6	7	8	9	10

Heap

▶ 최대 힙과 최소 힙

1) 최대 힙 (Max Heap) : 루트에 최대값이 저장된다.

부모의 키 값 \geq 자식의 키 값



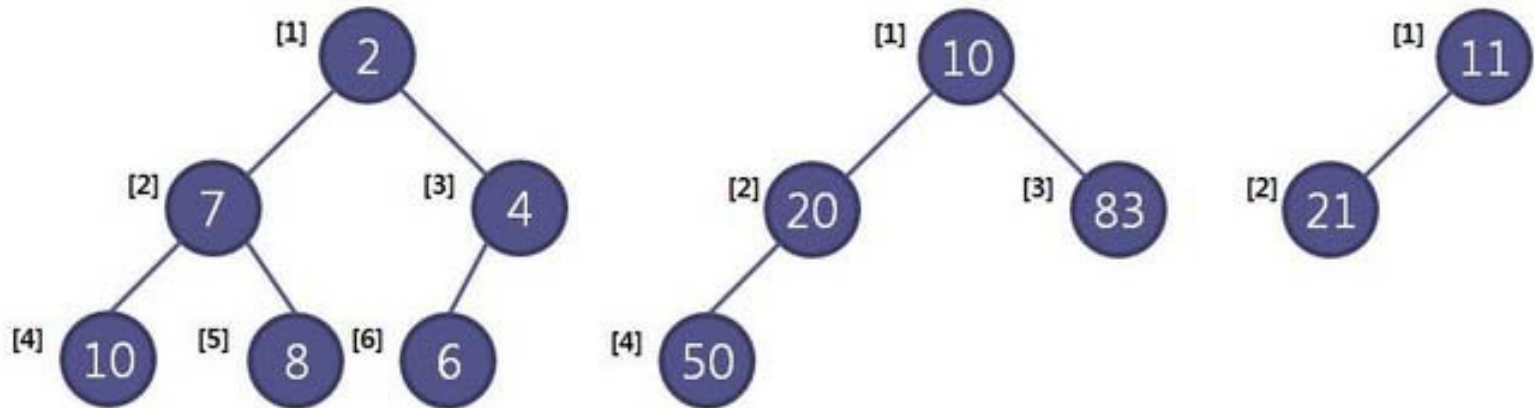
[그림 1] 최대 힙의 예

Heap

▶ 최대 힙과 최소 힙

2) 최소 힙 (Min Heap) : 루트에 최소값이 저장된다.

부모의 키 값 \leq 자식의 키 값



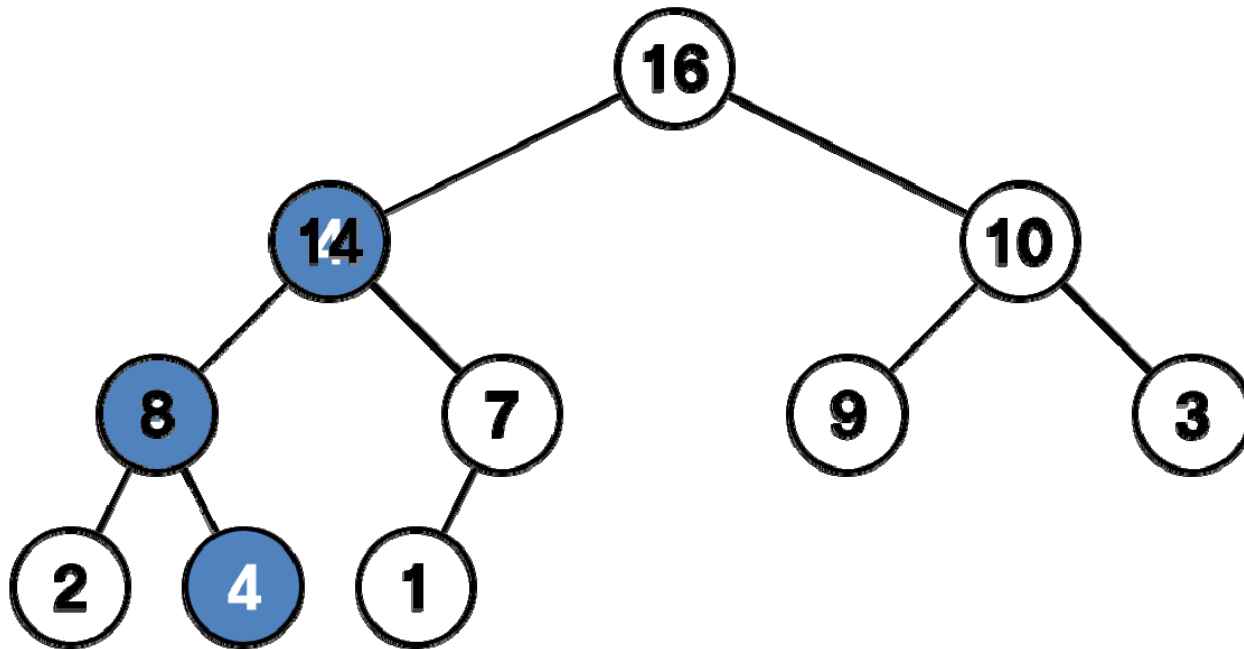
[그림 2] 최소 힙의 예

Heap

▶ 힙의 구현

MAX-HEAPIFY

특정 노드를 기점으로, 트리를 내려가며 최대 힙이 되도록 함



Heap

▶ 힙의 구현

MAX-HEAPIFY(A, i)

$L \leftarrow \text{LEFT-CHILD}(i)$ R

$\leftarrow \text{RIGHT-CHILD}(i)$

if $L \leq \text{heap_size}[A]$ and $A[L] > A[i]$

then $\text{largest} \leftarrow L$

else $\text{largest} \leftarrow i$

if $R \leq \text{heap_size}[A]$ and $A[R] > A[\text{largest}]$

then $\text{largest} \leftarrow R$

if $\text{largest} \neq i$

then $A[i] \leftrightarrow A[\text{largest}]$

MAX-HEAPIFY($A, \text{largest}$)

Heap

▶ 힙의 구현

BUILD-MAX-HEAP(A)

입력 받은 배열 A 를 최대 힙으로 만드는 함수.

자식을 갖는 마지막 노드부터 루트 노드까지 순서대로 검사하며 MAX-HEAPIFY 함수를 실행한다.

BUILD-MAX-HEAP(A)

$heap_size[A] \leftarrow length[A]$

for $i \leftarrow \lfloor length[A]/2 \rfloor$ downto 1

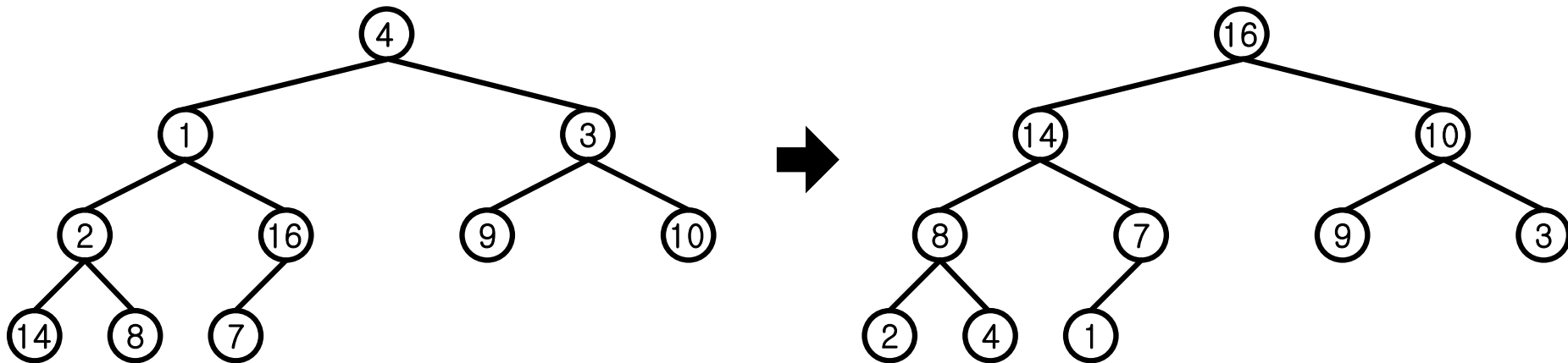
do MAX-HEAPIFY(A, i)

Heap

▶ 힙의 구현

BUILD-MAX-HEAP(A)

4	1	3	2	16	9	10	14	8	7
1	2	3	4	5	6	7	8	9	10



Priority Queue

▶ Priority Queue란?

힙 자료구조를 이용한 효율적인 큐(Queue) 알고리즘.

노드에 저장되는 키 값을 우선순위로 하고
우선순위가 최대/최소인 작업을 먼저 꺼내서 처리하는 방법으로,

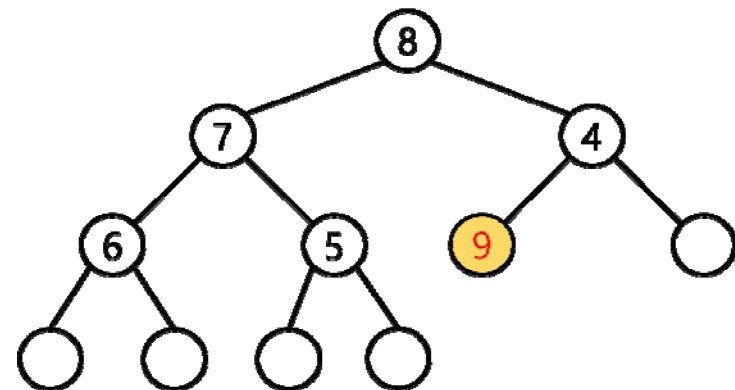
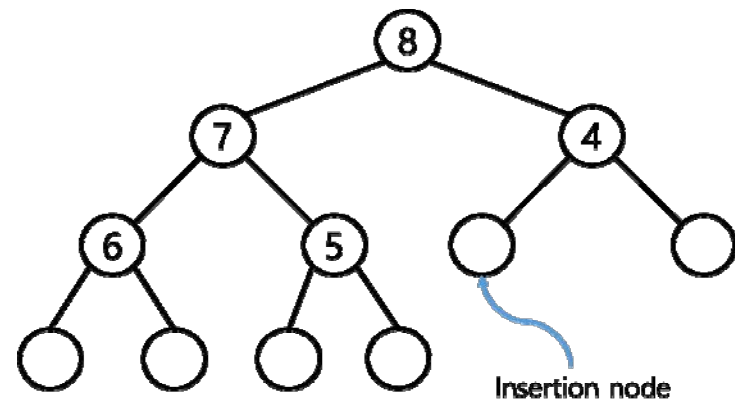
작업 스케줄링, 네트워크 트래픽 제어 등에 활용된다.

힙 자료구조에 삽입, 추출 프로세스를 추가하는 것만으로
간단하게 구현할 수 있으며,
힙과 마찬가지로 최대/최소 우선 순위 큐로 나뉘어진다.

Priority Queue

▶ 최대 우선 순위 큐의 삽입 과정

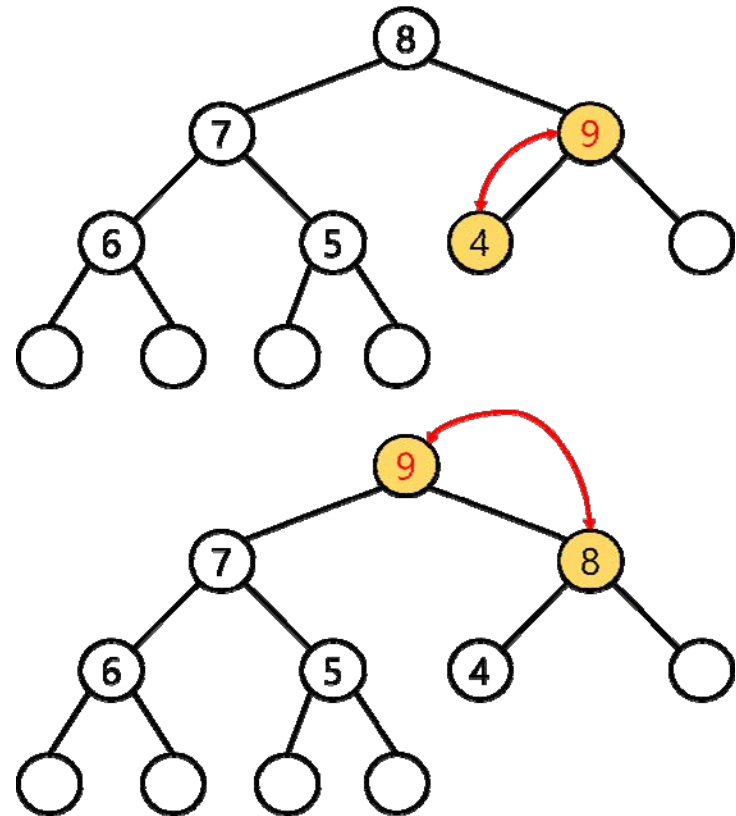
- 1) 트리의 가장 마지막 부분에 새로운 노드를 삽입한다.
- 2) 삽입한 노드와 부모 노드의 크기를 비교한다.
- 3) 부모 노드가 더 크면 삽입 프로세스를 종료한다.



Priority Queue

▶ 최대 우선 순위 큐의 삽입 과정

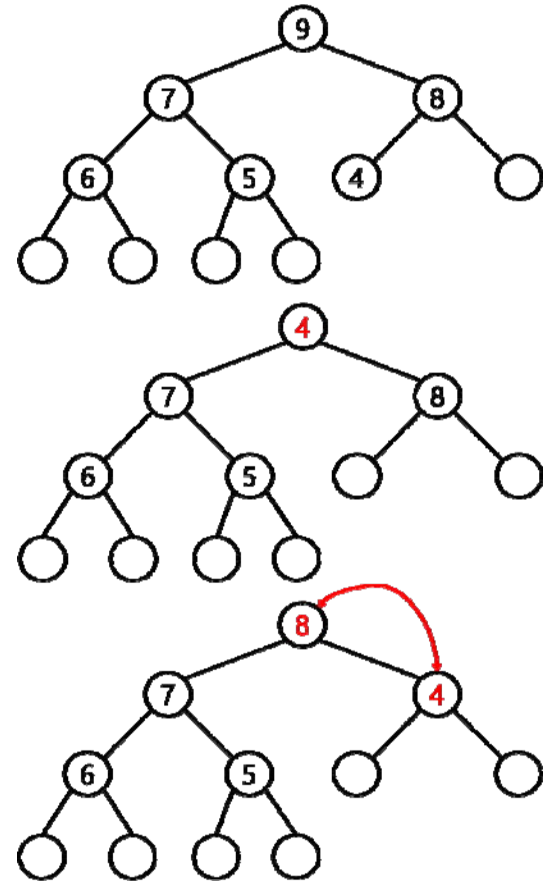
- 4) 삽입한 노드가 더 크면
이를 부모 노드와 교체한다.
- 5) 프로세스가 종료되거나 루트
에 도달할 때까지 상방향으로
연산을 반복한다.



Priority Queue

▶ 최대 우선 순위 큐의 추출 과정

- 1) 루트 노드를 추출한다.
- 2) 가장 마지막에 위치한 노드를 루트 위치로 이동시킨다.
- 3) 새로운 루트 노드를 기준으로 MAX-HEAPIFY를 수행한다.



Practice / Homework

data03.txt 파일의 데이터로 **Max Heap**을 만들고
삽입, 추출 프로세스를 추가하여 **Max Priority Queue**를 구현하라.
삽입, 추출을 수행할 때마다 남은 노드 목록을 출력하여야 한다.

- 1) 하나의 노드와, 이를 배열로 갖는 힙 구조체를 정의한다.
- 2) 힙을 생성하고 동적 공간을 할당한 후에,
파일 읽기가 수행되는 순서대로 각 데이터 정보를 저장한다.
- 3) MAX-HEAPIFY와 BUILD-MAX-HEAP을 구현 및 사용하여,
2번에서 얻은 배열(힙)을 Max Heap으로 바꾼다.

Practice / Homework

data03.txt 파일의 데이터로 **Max Heap**을 만들고
삽입, 추출 프로세스를 추가하여 **Max Priority Queue**를 구현하라.
삽입, 추출을 수행할 때마다 남은 노드 목록을 출력하여야 한다.

- 4) 삽입 함수 insert()와 추출 함수 extractMax()를 구현한다.
- 5) 완성된 프로그램을 구동하고, 파일을 큐에 읽어 들인 후에도
사용자가 노드를 추가하거나 우선도가 가장 높은 노드를 추출(삭제)할 수 있도록 메인 함수 main()을 완성한다.

Practice / Homework

data03.txt 파일의 데이터로 **Max Heap**을 만들고
삽입, 추출 프로세스를 추가하여 **Max Priority Queue**를 구현하라. 삽입,
추출을 수행할 때마다 남은 노드 목록을 출력하여야 한다. 아래
함수들은 전부 구현되어야 한다.

a. insert(S, x)

- S에 원소 x를 새로 넣는다.

b. max(S)

- S에서 키 값이 최대인 원소를 리턴한다.

c. extract_max(S)

- S에서 키 값이 최대인 원소를 제거한다.

d. increase_key(S, x, k)

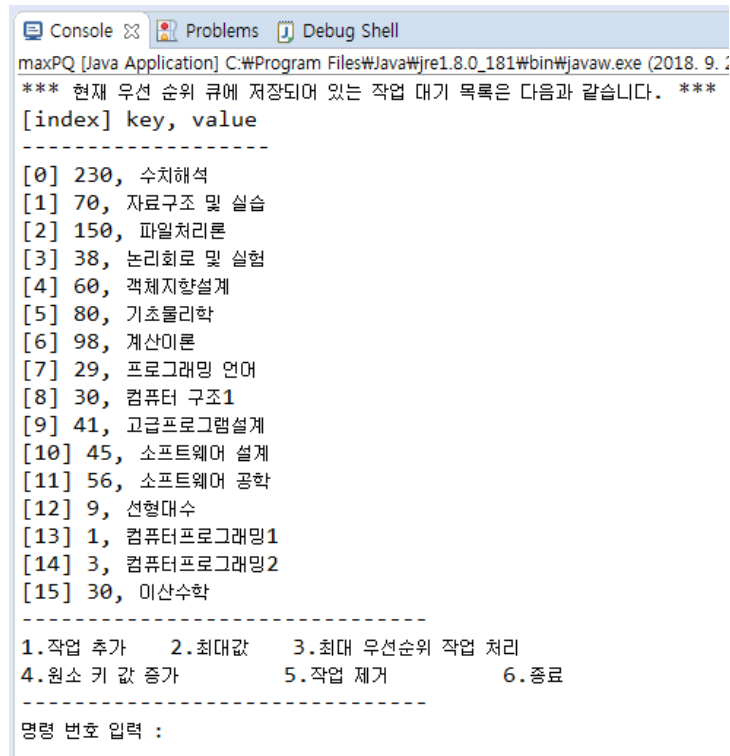
- 원소 x의 키 값을 k로 증가시킨다. 이 때 k는 x의 현재 키 값보다 작아지지 않는다고 가정한다

e. h_delete(S, x)

- S에서 노드 x를 제거한다. 제거 후 Max heap 유지

Practice / Homework

▶ 결과 화면 예시 : sample03_MaxPriorityQueue.exe



```
maxPQ [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018. 9. 2)
*** 현재 우선 순위 큐에 저장되어 있는 작업 대기 목록은 다음과 같습니다. ***
[index] key, value
-----
[0] 230, 수치해석
[1] 70, 자료구조 및 실습
[2] 150, 파일처리론
[3] 38, 논리회로 및 실험
[4] 60, 객체지향설계
[5] 80, 기초물리학
[6] 98, 계산이론
[7] 29, 프로그래밍 언어
[8] 30, 컴퓨터 구조1
[9] 41, 고급프로그래밍설계
[10] 45, 소프트웨어 설계
[11] 56, 소프트웨어 공학
[12] 9, 선형대수
[13] 1, 컴퓨터프로그래밍1
[14] 3, 컴퓨터프로그래밍2
[15] 30, 이산수학
-----
1.작업 추가    2.최대값    3.최대 우선순위 작업 처리
4.원소 키 값 증가    5.작업 제거    6.종료
-----
명령 번호 입력 :
```

- 명령어 입력에 따라 작업 수행한 결과를 화면에 출력

Practice / Homework

※ 보고서 작성 양식

- a . 알고리즘 설명
 - 실행시간 분석, 장 · 단점

- b . 결과 화면

- c . 표지 제외 1~2 페이지 이내로 작성

Practice / Homework

※ 그 외 실습 과제 수행 중 유의 사항

- a . 과제 제출은 e-mail로, 프로젝트 파일을 폴더 짝으로 zip으로 압축하여 보낼 것.
- b . 메일 제목과 zip 파일명은 항상 아래의 양식으로 작성.
>> **[알고리즘]01_주차_학번_이름**
- c . **과제 평가는 별도의 input data를 사용**함. (양식은 동일)

Practice / Homework

과제 제출 안내	
제출 방법	E-mail (<u>ghmin@cnu.ac.kr</u>)
메일 제목	[알고리즘]01_주차_학번_이름_maxPQ
File 이름	[알고리즘]01_주차_학번_이름_maxPQ.zip
제출 기한	10월 1일 (월) 23시 59분 59초 까지

과제 평가 감점 사항	
제출 지연 (수업 시작부터)	- 50% / 1주
요구 사항 누락 / 결과값 불일치	- 10 ~ 20% / 1개
코드 Error	- 50 ~ 100%
과제 Copy	0점