

1. (최대힙 구현, binary max-heap) 다음은 최대힙의 삽입, 삭제 동작을 구현한 코드이다. 아래 코드를 입력하고 실행하면서 힙의 구현법을 익히시오. (Reference: https://en.wikipedia.org/wiki/Binary_heap, CC-BY-SA, <https://algs4.cs.princeton.edu/24pq/MaxPQ.java.html>, GPLv3)

```
public class Test {
    public static void main(String[] args) {
        int v[]={4, 2, 9, 5, 7, 5, 8, 10, 15};
        SimpleHeap heap=new SimpleHeap();
        for (int i = 0; i < v.length; i++) {
            heap.add(v[i]);
            System.out.println(heap);
        }
        for (int i = 0; i < v.length; i++) {
            System.out.println(heap.remove()+"=>" + heap);
        }
    }
}

public class SimpleHeap {
    int last=-1, MaxHeapSize=4;
    int heap[]=new int[MaxHeapSize];
    private void resize() {
        MaxHeapSize*=2;
        heap=Arrays.copyOf(heap, MaxHeapSize);
    }
    private void swap(int m, int n) {
        int temp=heap[m];
        heap[m]=heap[n];
        heap[n]=temp;
    }
    public void add(Integer data) {
        if(last+1==MaxHeapSize) resize();
        heap[++last]=data; // heap의 마지막 노드 다음 위치에 새 자료 삽입
        for (int child=last; child>0; ) { // 마지막 노드를 heapify-up
            int parent=(child-1)/2;
            if(heap[child]<=heap[parent]) break; // 최대힙조건 검사
            swap(child, parent); // 조건 불만족 시 교환
            child=parent;
        }
    }
    public int remove() {
        if(last<0) throw new RuntimeException("heap empty");
        int data=heap[0]; // root 노드 자료 추출
        heap[0]=heap[last--]; // 마지막 노드 root로 이동 & 크기 1 감소
        for (int parent=0, child=2*parent+1; child<=last; parent=child, child=2*parent+1) { // heapify-down
            if(child<last && heap[child]<heap[child+1]) child++;
            if(heap[child]<=heap[parent]) break;
            swap(child,parent);
        }
        return data;
    }
    @Override
    public String toString() {
        return Arrays.toString(heap);
    }
}
```

2. (우선순위큐, 최소힙, 자바클래스 활용) 다음은 자바클래스 PriorityQueue(우선순위큐)를 활용한 예시 코드이다. PriorityQueue는 priority heap을 구현한 자바클래스이다. 아래 코드를 입력하고 실행하면서 자바에서 우선순위큐의 사용법을 익히시오. (삽입: add(), 큐 root 값 확인: peek(), 큐 root 값 삭제: remove(), 우선순위큐 크기: size())

```
public class Test {  
    public static void main(String[] args) {  
        int v[]={4, 2, 9, 5, 7, 5, 8, 10, 15};  
        PriorityQueue<Integer> minheap=new PriorityQueue<>(); // 최소힙  
        for (int i = 0; i < v.length; i++) {  
            minheap.add(v[i]); // 최소힙에 삽입  
            System.out.println(minheap);  
        }  
        System.out.println("최소값="+minheap.peek()); // 삭제하지 않고 큐의 root값 반환  
        while(minheap.size()>0) {  
            System.out.println(minheap.remove()+"=>" +minheap);  
        }  
    }  
}
```

3. (우선순위큐, 최대힙, 자바클래스 활용) 다음은 자바클래스 PriorityQueue(우선순위큐)를 활용한 예시 코드이다. PriorityQueue는 priority heap을 구현한 자바클래스이다. 아래 코드를 입력하고 실행하면서 자바에서 우선순위큐의 사용법을 익히시오. (삽입: add(), 큐 root 값 확인: peek(), 큐 root 값 삭제: remove(), 우선순위큐 크기: size())

```
public class Test {  
    public static void main(String[] args) {  
        int v[]={4, 2, 9, 5, 7, 5, 8, 10, 15};  
        PriorityQueue<Integer> maxheap=new PriorityQueue<>(Collections.reverseOrder()); // 최대힙  
        for (int i = 0; i < v.length; i++) {  
            maxheap.add(v[i]);  
            System.out.println(maxheap);  
        }  
        System.out.println("최대값="+maxheap.peek()); // 삭제하지 않고 큐의 root값 반환  
        while(maxheap.size()>0) {  
            System.out.println(maxheap.remove()+"=>" +maxheap);  
        }  
    }  
}
```

4. (실습: 가장 작은 k개 정수 출력) 다음은 배열 v에 저장된 N개 임의 정수 중 크기 순으로 가장 작은 k개 정수를 출력하는 코드이다. 아래 코드를 heap을 이용하여 구현하시오. 구현된 방법의 시간복잡도는 얼마인가? 더 빠른 방법이 있는가?

```
public class Test {  
    public static void main(String[] args) {  
        Random random=new Random();  
        int N=1000000;  
        int v[]=new int[N];  
        int k=10;  
        for (int i = 0; i < N; i++) v[i]=random.nextInt(N);  
  
    }  
}
```

References

- C로 쓴 자료구조론 (Fundamentals of Data Structures in C, Horowitz et al.). 이석호 역. 사이텍미디어. 1993.
- 쉽게 배우는 알고리즘: 관계 중심의 사고법. 문병로. 한빛아카데미. 2013.
- C언어로 쉽게 풀어 쓴 자료구조. 천인국 외 2인. 생능출판사. 2017.
- 김윤명. (2008). 뇌를 자극하는 Java 프로그래밍. 한빛미디어.
- 남궁성. 자바의 정석. 도우출판.
- 김윤명. (2010). 뇌를 자극하는 JSP & Servlet. 한빛미디어.