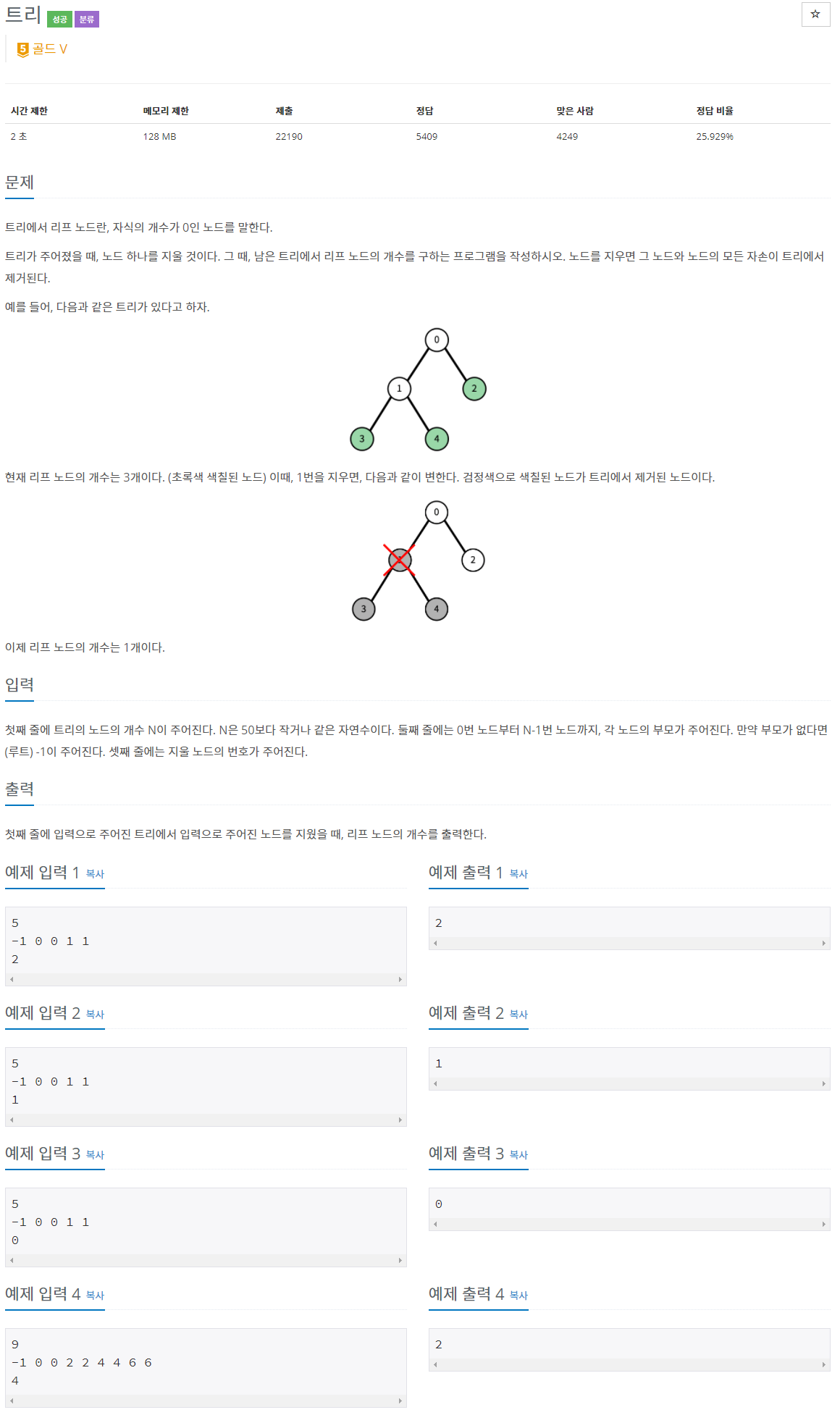
1068 (자료구조)



부모를 입력 받고, 노드를 삭제한 후 리프 노드를 count하는 문제였음

노드를 삭제할 때에는 recursion을 이용해 현재 노드의 부모 인덱스가 삭제된 노드라면 연쇄적으로 삭제가 일어나도록 구현해 줌

리프노드를 카운트 할 때에는 깊이 우선 탐색(DFS)을 활용하여 현재 노드를 부모로 가지는 노드가 하나라도 존재한다면 자식 노드를 연쇄적으로 탐색하도록 함, 재귀 함수 속에서 자식 노드가 하나라도 존재하는지를 기록해 주기 위해 boolean 타입의 변수를 하나 생성하여 자식 노드가 없다면 리프노드의 카운트를 증가함

1927 (자료구조)

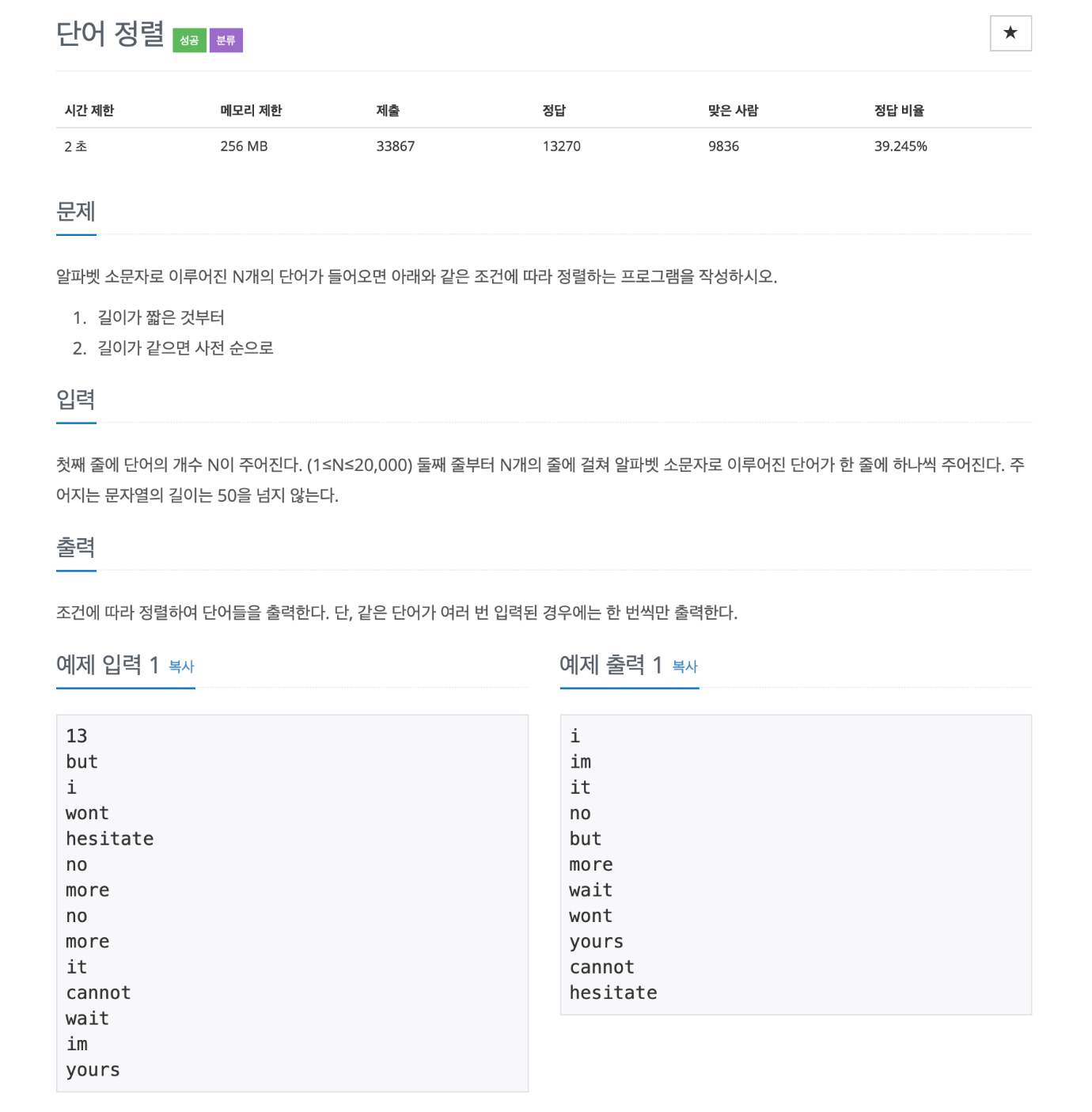


일반적인 queue 는 FIFO 구조를 가지지만, 우선순위 queue는 내부에서 우선순위를 자동으로 정해, 들어온 순서가 아닌 정한 우선순위가 높은 것부터 먼저 내보내는 방식을 가짐

내부 구조가 힙으로 구성되어 있으며 (정확히 우선순위 queue는 min heap 자료구조 이용), 시간 복잡도는 O(nlogn)이다.

java에서 제공하는 라이브러리인 PriorityQueue를 통해 선언하여 생각보다 쉽게 풀 수 있었음

1181 (정렬)



1. 단어 길이 순으로 정렬한 뒤, 길이가 같을 경우 사전순으로 정렬해야 함
2. 중복되는 단어가 있을 경우 한 번만 출력

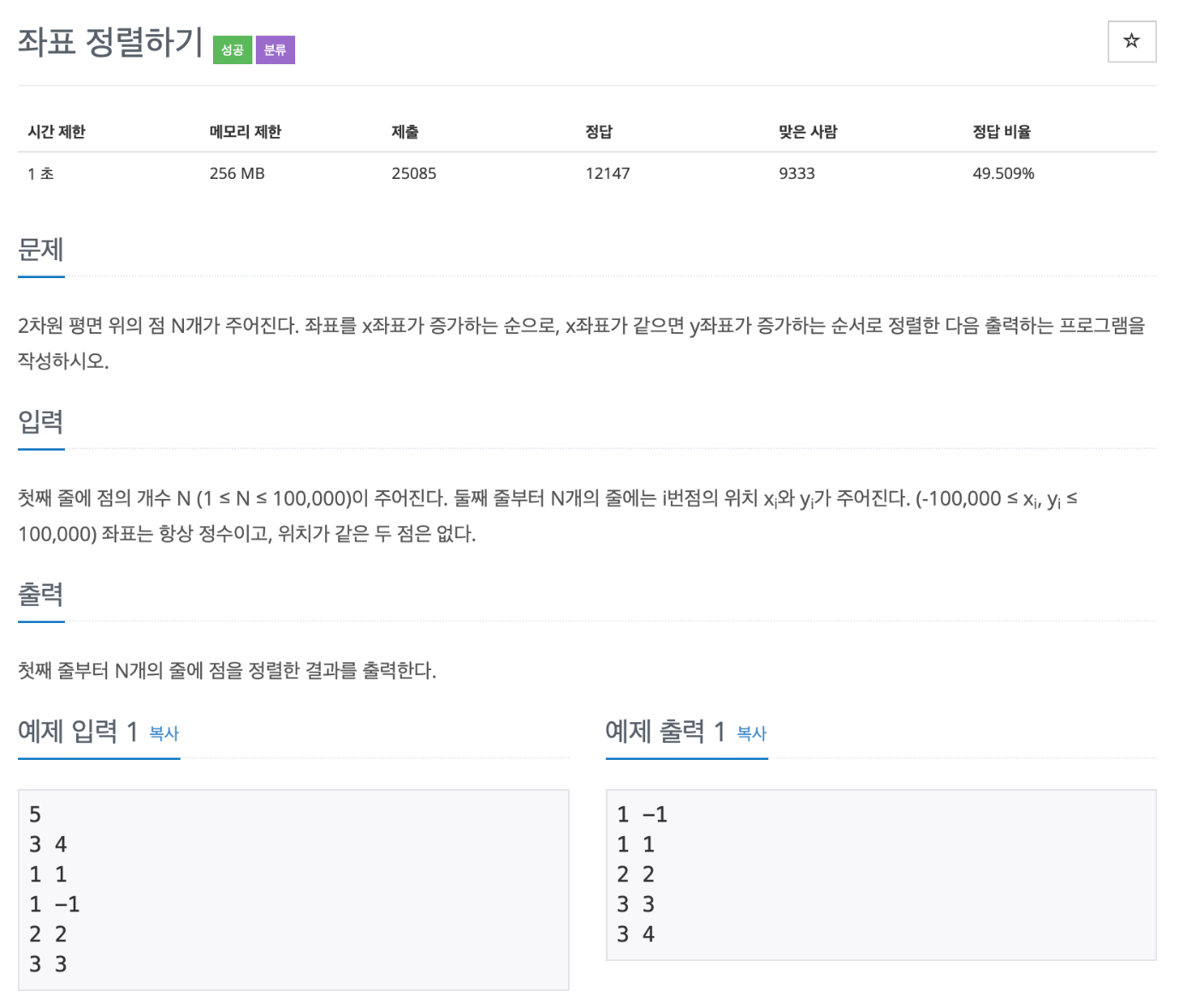
이 2가지를 주의해야 함

* 배열을 특정한 규칙에 의해 정렬하고 싶은 경우 Arrays.sort 메소드에 Comparator를 구현하면 됨
* Arrays.sort(arr, new Comparator<String>() {  
   public int compare(String o1, String o2) {  
   if(o1.length() == o2.length()) {  
   return o1.compareTo(o2);  
   }  
    
   else {  
   return o1.length() - o2.length();  
   }  
   }  
  });

이런 식으로 단어 사전 순 정렬은 compareTo 메소드 써주면 됨

Scanner 를 통해 입력 받았고, Arrays.sort 를 이용하여 배열을 비교하여 정렬하였음

11650 (정렬)



좌표 정렬 문제이므로, Arrays.sort()에서 Comparator 인터페이스 사용하는 방식으로 코딩함

BufferedReader 를 이용하여 데이터 입력 받고, StringBuilder 이용해서 추가하여 출력함

Arrays.sort(arr, new Comparator<int[]>() {  
 public int compare(int[]o1, int[]o2) {  
 if(o1[0] == o2[0]) return o1[1] - o2[1];  
 else return o1[0] - o2[0];  
 }  
 });  
  
 for(int i = 0; i < n; i++) {  
 sb.append(arr[i][0]).append(" ").append(arr[i][1]).append("\n");  
 }  
 System.out.println(sb);  
}