

## Group Activity 04

(3인 혹은 4인으로 팀을 구성하여 아래의 문제를 푼다. 팀 구성은 매 시간마다 달라져도 된다.)

팀원1: \_\_\_\_\_

팀원2: \_\_\_\_\_

팀원3: \_\_\_\_\_

팀원4: \_\_\_\_\_

1. 삽입정렬(insertion sort)에서 데이터를 삽입할 위치를 이진검색(binary search) 알고리즘으로 찾을 수도 있다. 그렇게 한다면 삽입정렬 알고리즘의 최악의 경우 시간복잡도는? 이유는?

2. 합병(merge) 정렬 알고리즘에서 데이터를 절반으로 나누지 않고 항상 1/3과 2/3로 나눈다면 최악의 경우 시간 복잡도는? 이유는?

3.  $n$ 개의 정수의 수열  $a_1, a_2, \dots, a_n$ 에 대해서  $i < j$ 이면서  $a_i > a_j$ 일 때  $(a_i, a_j)$ 는 역전된(inverted) 쌍이라고 부른다. 입력에서 역전된 쌍의 개수가 최대  $n$ 개를 넘지 않는다고 가정하면 삽입정렬 알고리즘의 최악의 경우 시간 복잡도는? 이유는?

4.  $n$ 개의 정수들 중에서  $n/4$ 번째로 작은 정수를  $O(n)$  시간에 찾는 알고리즘이 존재한다. 이 알고리즘을 이용하여 항상  $n/4$ 번째로 작은 정수를 pivot으로 선택하여 사용한다면 quick sort 알고리즘의 최악의 경우 시간복잡도는? 이유는?

5. 안정된 정렬(stable sort) 알고리즘이란 동일한 값들은 입력된 순서대로 출력되는 정렬 알고리즘을 말한다. 다음은 강의 슬라이드에 있는 선택 정렬(selection sort) 알고리즘이다. 이 알고리즘은 안정된 정렬 알고리즘인가? 만약 아니라면 안정된 정렬 알고리즘이 되도록 수정하라. 코드를 작성할 필요는 없고 적절하게 말로 설명하면 된다.

```
selectionSort(A[], n)  ▷ 배열 A[1...n]을 정렬한다.
{
    for last ← n downto 2  {
        A[1...last]중 가장 큰 수 A[k]를 찾는다;
        A[k]와 A[last]의 값을 교환한다;
    }
}
```

6. 강의 슬라이드에 기술된 버블정렬과 삽입정렬은 안정된 정렬 알고리즘인가? 만약 아니거나 혹은 불분명하다면 어떻게 해야 안정된 정렬 알고리즘이 되는지 설명하라.

7. 합병 정렬 알고리즘이 안정된 정렬이 되도록 합병(merge) 함수를 pseudocode의 형태로 작성하라.

```
merge(A[], p, q, r) {
```

```
}
```

8. 슬라이드 31페이지와 35페이지에 있는 빠른 정렬 알고리즘은 안정된 정렬 알고리즘인가? 만약 피벗 선택을 슬라이드 41페이지에 설명한 Median of Three 혹은 랜덤으로 피벗을 선택하면 어떻게 되는가?