

7.1 다음 중 최선의 입력과 최악의 입력에 대한 선택 정렬의 시간 복잡도는?

(1) 최선: $O(n)$, 최악: $O(n^2)$

(2) 최선: $O(n \log n)$, 최악: $O(n^2)$

(3) 최선: $O(n^2)$, 최악: $O(n^2)$

(4) 최선: $O(n^2)$, 최악: $O(n^3)$

7.2 다음 중 최선의 입력과 최악의 입력에 대한 삽입 정렬의 시간 복잡도는?

(1) 최선: $O(n)$, 최악: $O(n^2)$

(2) 최선: $O(n \log n)$, 최악: $O(n^2)$

(3) 최선: $O(n^2)$, 최악: $O(n^2)$

(4) 최선: $O(n^2)$, 최악: $O(n^3)$

7.3 다음 중 정렬되지 않은 배열로 집합을 구현하는 경우에 비해 정렬된 배열을 더 효율적인 알고리즘이 가능한 집합의 연산이 아닌 것은?

(1) 삽입 연산

(2) 집합의 비교

(3) 합집합

(4) 차집합

7.4 다음 중 순차 탐색에 대한 설명으로 옳은 것은?

(1) 정렬되지 않은 테이블에서도 동작한다.

(2) 최선과 최악의 입력에 대해 시간 복잡도 차이가 없다.

(3) 리스트를 균등하게 분할하여 탐색한다.

(4) 탐색 값과 위치가 비례한다고 가정한다.

7.5 다음 중 이진 탐색의 응용 분야로 적절하지 않은 경우는?

(1) 효율적인 탐색이 필요할 때

(2) 테이블이 정렬되어 있을 때

(3) 데이터의 입력과 삭제가 빈번할 때

(4) 탐색이 매우 빈번하게 발생할

7.6 다음 중 보간 탐색에 대한 설명으로 적절하지 않은 것은?

(1) 이진 탐색의 개선된 알고리즘이다

(2) 탐색기가 존재할 위치를 예측하여 탐색한다.

(3) 리스트를 균등하게 분할하여 탐색한다.

(4) 탐색 값과 위치가 비례한다고 가정한다.

7.7 다음 중 좋은 해시 함수의 조건이 아닌 것은?

- ① 코드가 간단해야 한다.
- ② 충돌이 적어야 한다.
- ③ 주소가 테이블에 고르게 분포되어야 한다.
- ④ 계산이 빨라야 한다.

7.8 다음은 해싱에서의 오버플로 처리 방법들이다. 나머지와 다른 하나는?

- ① 선형 조사법 ② 이차 조사법 ③ 이중 해싱법 ④ 체이닝

7.9 다음은 항목을 오름차순으로 정렬하는 삽입 정렬 함수이다. 빈 칸을 채워라.

```
def insertion_sort(A) :
```

```
    n = len(A)
```

```
    for i in range(1, n) :
```

```
        key = A[i]
```

```
        j = i-1
```

```
        while j >= 0 and  $A[j] > \text{key}$  :
```

```
             $A[j+1] = A[j]$ 
```

```
            j -= 1
```

```
        A[j+1] = key
```

7.10 다음의 정렬기법을 이용하여 정수 배열을 오름차순으로 정렬하라. 각 단계에서의 배열의 내용을 나타내어라.

7	4	9	6	3	8	7	5
---	---	---	---	---	---	---	---

- (1) 선택 정렬
- (2) 삽입 정렬
- (3) 버블 정렬

7.11 삽입 정렬이 안정성을 만족하는 이유를 설명해 보라.

7.12 삽입 정렬을 위한 최선의 입력과 최악의 입력을 시간 복잡도를 이용해 설명하라.

최선 입력 $O(n)$: 이미 정렬 되어 있는 경우 $n-1$ 번
최악 입력 $O(n^2)$: 역순으로 정렬된 경우