

11 장 정렬고급 연습문제

1 버블 정렬에 대한 설명으로 거리가 먼 것은? 4

- ① 첫 번째 값부터 시작해서 바로 앞뒤 데이터를 비교하며 큰 것을 뒤로 보내는 방법을 사용한다.
- ② 키 순서, 이름 순서, 몸무게 순서 등을 정렬할 때 사용할 수 있다.
- ③ 정렬은 오름차순과 내림차순으로 정렬할 수 있다.
- ④ 삽입 정렬이나 선택 정렬보다 성능이 나쁘다.

2 오름차순으로 정렬하는 버블 정렬 함수의 ❶~❷에 적합한 코드를 다음 중 고르시오.

❶ `range(n-1, 0, -1)` ❷ `ary[cur] > ary[cur+1]`

```
range(n-1, 0, -1) range(n, 0, -1) range(0, n-1, 1) range(0, n, 1)
ary[cur+1] > ary[cur+2] ary[cur] > ary[cur+1] ary[cur] < ary[cur+1]
ary[cur+1] > ary[cur+2]
```

```
def BubbleSort(ary) :
    n = len(ary)
    for end in ❶ :
        for cur in range(0, end) :
            if (❷) :
                ary[cur], ary[cur+1] = ary[cur+1], ary[cur]
    return ary
```

3 버블 정렬이 선택 정렬이나 삽입 정렬보다 성능이 더 좋은 경우는? 1

- ① 데이터가 이미 정렬이 거의 되어 있고, 중간에 정렬되지 않은 데이터가 하나 있을 때
- ② 데이터가 정렬이 전혀 안 되어 있는 경우
- ③ 오름차순 정렬할 때
- ④ 내림차순 정렬할 때

4 퀵 정렬에 대한 설명으로 거리가 먼 것은? 4

- ① 기준을 하나 뽑은 후 기준보다 작은 그룹과 큰 그룹을 나누어 다시 각 그룹으로 정렬한다.
- ② 키 순서, 이름 순서, 몸무게 순서 등을 정렬할 때 사용할 수 있다.
- ③ 정렬은 오름차순과 내림차순으로 정렬할 수 있다.
- ④ 삽입 정렬이나 선택 정렬보다 성능이 더 나쁘다.

5 오름차순으로 정렬하는 버블 정렬 함수의 ❶~❷에 적합한 코드를 다음 중 고르시오. 단 중간에 정렬이 완료되면 종료되는 기능이 추가되었다.

❶ changeYN = True ❷ not changeYN

```
changeYN = True changeYN = False changeYN = Yes changeYN = No
not changeYN changeYN changeYN == Yes changeYN == No
```

```
def BubbleSort(ary) :
    n = len(ary)
    for end in range(n-1, 0, -1) :
        changeYN = False
        for cur in range(0, end) :
            if (ary[cur] > ary[cur+1]) :
                ary[cur], ary[cur+1] = ary[cur+1], ary[cur]
                ❶
        if ❷ :
            break
    return ary
```

6 오름차순으로 정렬하는 퀵 정렬 함수 ❶~❷에 적합한 코드를 다음 중 고르시오.

❶ ary[n // 2] ❷ quickSort(leftAry) + [pivot] + quickSort(rightAry)

```
ary[n] ary[n // 2] ary[n / 2] ary[n* 2]
quickSort(leftAry) + [pivot] + quickSort(rightAry)
[pivot] + quickSort(rightAry) + quickSort(leftAry)
quickSort(leftAry) + quickSort(rightAry) + [pivot]
```

```
def quickSort(ary) :
    n = len(ary)
    if n <= 1 :
        return ary
    pivot = ❶
    leftAry, rightAry = [], []
    for num in ary :
        if num < pivot :
            leftAry.append(num)
        elif num > pivot :
            rightAry.append(num)
    return ❷
```

