<ol> <li>In Bubble Sort, how are elements compared and sorted?</li> <li>A. By comparing adjacent elements and swapping if necessary</li> <li>By selecting the smallest element and moving it to the start</li> <li>By dividing the list into two halves and sorting each half</li> <li>By moving the largest element to the end in each iteration</li> </ol>
2. In Python, how can Bubble Sort be improved to stop early if the array is already sorted?
<ul> <li>A. Using a break statement when no swaps occur in an iteration</li> <li>B. Decreasing the number of iterations by half</li> <li>C. Sorting the array in reverse order first</li> <li>D. Using a different sorting algorithm</li> </ul>
3. What is the space complexity of Bubble Sort?
A. O(n) B. O(n log n) C. O(n^2) D. O(1)
4. How many times does the outer loop of Bubble Sort run for an array of size 'n'?
A. n B. n-1 C. n/2 D. 2n
5. For the given bubble sort algorithm:
If we want to sort the elements ["apple", "banana", "orange", "kiwi"], what should be the value of blank?

```
for i in range(size-1):
    swapped = False
    for j in range(blank):
        if elements[j] > elements[j+1]:
            tmp = elements[j]
            elements[j] = elements[j+1]
            elements[j+1] = tmp
            swapped = True

if not swapped:
        break
```

- A. Size-1-i
- B. size-i
- C. size
- D. i