

Consider the following algorithm:

```
def order(L):  
    """ (list of numbers) -> None  
        Order L from smallest to largest. L is changed in-place. """  
    i = 1  
    while i < len(L):  
        j = i  
        while j > 0 and L[j] < L[j-1]:  
            L[j], L[j-1] = L[j-1], L[j] # swap L[j] and L[j-1]  
            j = j - 1  
        i = i + 1
```

1. Compute the *exact* number of “swaps” (executing the line that says swap) performed by the algorithm in the worst-case, on any list  $L$  of length  $n$ .

2. Compute the *exact* number of “steps” (basic operations) performed by the algorithm in the worst-case, on any list  $L$  of length  $n$ .