

Leandro Lopez

1)

Our problem instances

A1 = [63, 44, 17, 77, 20, 6, 99, 84, 52, 39]

A2 = [84, 52, 39, 6, 20, 17, 77, 99, 63, 44]

A3 = [99, 84, 77, 63, 52, 44, 39, 20, 17, 6]

Input: An array A and indices i and j.

Output: An array where A[i] and A[j] have been swapped.

def Swap(A, i, j):

temp = A[i]

A[i] = A[j]

A[j] = temp

Input: An array A

Output: Our array sorted in Ascending order

def SelectionSort(A):

we begin iterating through our array, from the tail to head

for i in range(len(A)-1, 0, -1):

m = i

comparisons = 0

swaps = 0

we iterate through the index we will compare to our initial index, I

and proceed to compare, if j is larger than our previously instantiated largest variable m

j becomes the new largest

for j in range(0, i):

comparisons += 1

if A[j] > A[m]:

m = j

if our current index of i does not equal the largest value, m, we swap the values

if m != i:

Swap(A, i, m)

swaps += 1

print(f"Iteration {len(A) - i}: {A}")

Comparisons: {comparisons}

Swaps: {swaps}")

print statements for formatting

print("\n*****Result 1*****")

result1 = SelectionSort(A1)

print("=====")

print("\n*****Result 2*****")

result2 = SelectionSort(A2)

print("=====")

```
print("\n*****Result 3*****")
result3 = SelectionSort(A3)
print("=====")
```

```
*****Result 1*****
Iteration 1: [63, 44, 17, 77, 20, 6, 39, 84, 52, 99]
Comparisons: 9
Swaps: 1
Iteration 2: [63, 44, 17, 77, 20, 6, 39, 52, 84, 99]
Comparisons: 8
Swaps: 1
Iteration 3: [63, 44, 17, 52, 20, 6, 39, 77, 84, 99]
Comparisons: 7
Swaps: 1
Iteration 4: [39, 44, 17, 52, 20, 6, 63, 77, 84, 99]
Comparisons: 6
Swaps: 1
Iteration 5: [39, 44, 17, 6, 20, 52, 63, 77, 84, 99]
Comparisons: 5
Swaps: 1
Iteration 6: [39, 20, 17, 6, 44, 52, 63, 77, 84, 99]
Comparisons: 4
Swaps: 1
Iteration 7: [6, 20, 17, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 3
Swaps: 1
Iteration 8: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 2
Swaps: 1
Iteration 9: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 1
Swaps: 0
=====
```

```
*****Result 2*****
Iteration 1: [84, 52, 39, 6, 20, 17, 77, 44, 63, 99]
Comparisons: 9
Swaps: 1
Iteration 2: [63, 52, 39, 6, 20, 17, 77, 44, 84, 99]
Comparisons: 8
Swaps: 1
Iteration 3: [63, 52, 39, 6, 20, 17, 44, 77, 84, 99]
Comparisons: 7
Swaps: 1
Iteration 4: [44, 52, 39, 6, 20, 17, 63, 77, 84, 99]
Comparisons: 6
Swaps: 1
Iteration 5: [44, 17, 39, 6, 20, 52, 63, 77, 84, 99]
Comparisons: 5
```

Swaps: 1
 Iteration 6: [20, 17, 39, 6, 44, 52, 63, 77, 84, 99]
 Comparisons: 4
 Swaps: 1
 Iteration 7: [20, 17, 6, 39, 44, 52, 63, 77, 84, 99]
 Comparisons: 3
 Swaps: 1
 Iteration 8: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
 Comparisons: 2
 Swaps: 1
 Iteration 9: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
 Comparisons: 1
 Swaps: 0
 =====

*****Result 3*****

Iteration 1: [6, 84, 77, 63, 52, 44, 39, 20, 17, 99]
 Comparisons: 9
 Swaps: 1
 Iteration 2: [6, 17, 77, 63, 52, 44, 39, 20, 84, 99]
 Comparisons: 8
 Swaps: 1
 Iteration 3: [6, 17, 20, 63, 52, 44, 39, 77, 84, 99]
 Comparisons: 7
 Swaps: 1
 Iteration 4: [6, 17, 20, 39, 52, 44, 63, 77, 84, 99]
 Comparisons: 6
 Swaps: 1
 Iteration 5: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
 Comparisons: 5
 Swaps: 1
 Iteration 6: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
 Comparisons: 4
 Swaps: 0
 Iteration 7: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
 Comparisons: 3
 Swaps: 0
 Iteration 8: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
 Comparisons: 2
 Swaps: 0
 Iteration 9: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
 Comparisons: 1
 Swaps: 0
 =====

2)
 # Our problem instances
 A4 = [44, 63, 77, 17, 20, 99, 84, 6, 39, 52]

```
A5 = [52, 84, 6, 39, 20, 77, 17, 99, 44, 63]
A6 = [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
```

```
# Input: An array A of integers.
```

```
# Output: An array A sorted in increasing order.
```

```
def BubbleSort(A):
```

```
    total_comparisons = 0
```

```
    total_swaps = 0
```

```
    # we begin iterating through our array
```

```
    for i in range(len(A) - 1):
```

```
        comparisons = 0
```

```
        swaps = 0
```

```
        # we iterate through the index we will compare to our initial index, I
```

```
        # and proceed to compare, if j is than the value following j,
```

```
        # we swap the values
```

```
        for j in range(len(A) - i - 1):
```

```
            comparisons += 1
```

```
            if A[j + 1] < A[j]:
```

```
                Swap(A, j + 1, j)
```

```
                swaps += 1
```

```
        print(f"Iteration {i + 1}: {A}")
```

```
    Comparisons: {comparisons}
```

```
    Swaps: {swaps}")
```

```
    total_comparisons += comparisons
```

```
    total_swaps += swaps
```

```
    # if there are no swaps we quit
```

```
    if swaps == 0:
```

```
        break
```

```
    print(f"Total Comparisons: {total_comparisons}")
```

```
    Total Swaps: {total_swaps}")
```

```
# print statements for formatting
```

```
print("\n*****Result 4*****")
```

```
result4 = BubbleSort(A4)
```

```
print("=====")
```

```
print("\n*****Result 5*****")
```

```
result5 = BubbleSort(A5)
```

```
print("=====")
```

```
print("\n*****Result 6*****")
```

```
result6 = BubbleSort(A6)
```

```
print("=====")
```

```
*****Result 4*****
```

Iteration 1: [44, 63, 17, 20, 77, 84, 6, 39, 52, 99]
Comparisons: 9
Swaps: 6
Iteration 2: [44, 17, 20, 63, 77, 6, 39, 52, 84, 99]
Comparisons: 8
Swaps: 5
Iteration 3: [17, 20, 44, 63, 6, 39, 52, 77, 84, 99]
Comparisons: 7
Swaps: 5
Iteration 4: [17, 20, 44, 6, 39, 52, 63, 77, 84, 99]
Comparisons: 6
Swaps: 3
Iteration 5: [17, 20, 6, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 5
Swaps: 2
Iteration 6: [17, 6, 20, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 4
Swaps: 1
Iteration 7: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 3
Swaps: 1
Iteration 8: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 2
Swaps: 0
Total Comparisons: 44
Total Swaps: 23

=====

*****Result 5*****

Iteration 1: [52, 6, 39, 20, 77, 17, 84, 44, 63, 99]
Comparisons: 9
Swaps: 7
Iteration 2: [6, 39, 20, 52, 17, 77, 44, 63, 84, 99]
Comparisons: 8
Swaps: 6
Iteration 3: [6, 20, 39, 17, 52, 44, 63, 77, 84, 99]
Comparisons: 7
Swaps: 4
Iteration 4: [6, 20, 17, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 6
Swaps: 2
Iteration 5: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 5
Swaps: 1
Iteration 6: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]
Comparisons: 4
Swaps: 0
Total Comparisons: 39
Total Swaps: 20

=====

*****Result 6*****

Iteration 1: [6, 17, 20, 39, 44, 52, 63, 77, 84, 99]

Comparisons: 9

Swaps: 0

Total Comparisons: 9

Total Swaps: 0

=====

3)

A)

```
def power(x, p):  
    sum = x  
    for i in range(p-1):  
        sum *= x  
    return sum
```

result7 = power(2, 5) # result is 32

B) A7 = [12.3, 40.7, -9.1, 7.7, 6.4, 0, 8.9]

```
def evaluate(A, x):  
    n = len(A)  
    result = 0  
    for i in range(n):  
        result += A[i] * power(x, i)  
    return result
```

C) result8 = evaluate(A7, 5.4) # result is 227295.86317440012

D) The function uses a loop to iterate through each term, so the loop runs $n + 1$ times for a polynomial of n . Inside the loop, we call the power function of x raised to p power. So the total number of multiplications is: $1 + 2 + 3 + \dots + n + 1$.

We can evaluate that to $n(n+1)/2$, and this can be further evaluated to $(n+1)(n+2)/2$.

This falls under the worst case classification $O(n^2)$