Algorithm

Bubble Sort

Example

```
[5, 2, 8, 1, 10]
```

```
[2, 5, 8, 1, 10]
```

```
[2, 5, 1, 8, 10]
```

```
[2, 1, 5, 8, 10]
```

```
[1, 2, 5, 8, 10]
```

```
[5, 2, 8, 1, 10]
```

```python
def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
>>> bubble_sort([5, 2, 8, 1, 10])
========> Starting Bubble Sort

Initial list: [5, 2, 8, 1, 10]
List length: 5

-----> Outer Loop iteration #1

-> Inner Loop iteration #1
Left element: 5
Right element: 2
Not sorted: 5 > 2
Swapping...
Old list: [5, 2, 8, 1, 10]
New list: [2, 5, 8, 1, 10]

-> Inner Loop iteration #2
Left element: 5
Right element: 8
Already sorted: 5 < 8
No change: [2, 5, 8, 1, 10]
```

```
[2, 5, 8, 1, 10]
```

```python
def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
>>> bubble_sort([5, 2, 8, 1, 10])
========> Starting Bubble Sort

Initial list: [5, 2, 8, 1, 10]
List length: 5

-----> Outer Loop iteration #1

-> Inner Loop iteration #1
Left element: 5
Right element: 2
Not sorted: 5 > 2
Swapping...
Old list: [5, 2, 8, 1, 10]
New list: [2, 5, 8, 1, 10]

-> Inner Loop iteration #2
Left element: 5
Right element: 8
Already sorted: 5 < 8
No change: [2, 5, 8, 1, 10]
```

`[2, 5, 8, 1, 10]`

```python
def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
-> Inner Loop iteration #3
Left element: 8
Right element: 1
Not sorted: 8 > 1
Swapping...
Old list: [2, 5, 8, 1, 10]
New list: [2, 5, 1, 8, 10]

-> Inner Loop iteration #4
Left element: 8
Right element: 10
Already sorted: 8 < 10
No change: [2, 5, 1, 8, 10]
```

```
[2, 5, 1, 8, 10]
```

```python
def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
-> Inner Loop iteration #3
Left element: 8
Right element: 1
Not sorted: 8 > 1
Swapping...
Old list: [2, 5, 8, 1, 10]
New list: [2, 5, 1, 8, 10]

-> Inner Loop iteration #4
Left element: 8
Right element: 10
Already sorted: 8 < 10
No change: [2, 5, 1, 8, 10]
```

`[2, 5, 1, 8, 10]`

```python
def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
-----> Outer Loop iteration #2

-> Inner Loop iteration #1
Left element: 2
Right element: 5
Already sorted: 2 < 5
No change: [2, 5, 1, 8, 10]

-> Inner Loop iteration #2
Left element: 5
Right element: 1
Not sorted: 5 > 1
Swapping...
Old list: [2, 5, 1, 8, 10]
New list: [2, 1, 5, 8, 10]

-> Inner Loop iteration #3
Left element: 5
Right element: 8
Already sorted: 5 < 8
No change: [2, 1, 5, 8, 10]
```

```python
[2, 1, 5, 8, 10]

def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
-----> Outer Loop iteration #2

-> Inner Loop iteration #1
Left element: 2
Right element: 5
Already sorted: 2 < 5
No change: [2, 5, 1, 8, 10]

-> Inner Loop iteration #2
Left element: 5
Right element: 1
Not sorted: 5 > 1
Swapping...
Old list: [2, 5, 1, 8, 10]
New list: [2, 1, 5, 8, 10]

-> Inner Loop iteration #3
Left element: 5
Right element: 8
Already sorted: 5 < 8
No change: [2, 1, 5, 8, 10]
```

```
[2, 1, 5, 8, 10]

def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
-----> Outer Loop iteration #2

-> Inner Loop iteration #1
Left element: 2
Right element: 5
Already sorted: 2 < 5
No change: [2, 5, 1, 8, 10]

-> Inner Loop iteration #2
Left element: 5
Right element: 1
Not sorted: 5 > 1
Swapping...
Old list: [2, 5, 1, 8, 10]
New list: [2, 1, 5, 8, 10]

-> Inner Loop iteration #3
Left element: 5
Right element: 8
Already sorted: 5 < 8
No change: [2, 1, 5, 8, 10]
```

Python Searching and Sorting Algorithms: A Practical Approach

```
[2, 1, 5, 8, 10]
```

```python
def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
-----> Outer Loop iteration #3

-> Inner Loop iteration #1
Left element: 2
Right element: 1
Not sorted: 2 > 1
Swapping...
Old list: [2, 1, 5, 8, 10]
New list: [1, 2, 5, 8, 10]

-> Inner Loop iteration #2
Left element: 2
Right element: 5
Already sorted: 2 < 5
No change: [1, 2, 5, 8, 10]


-----> Outer Loop iteration #4

-> Inner Loop iteration #1
Left element: 1
Right element: 2
Already sorted: 1 < 2
No change: [1, 2, 5, 8, 10]

There was no need to swap! The list is now sorted
[1, 2, 5, 8, 10]
```

`[1, 2, 5, 8, 10]`

```python
def bubble_sort(lst):
    n = len(lst)

    for i in range(n):
        swapped = False

        for j in range(0, n-i-1):
            if lst[j] > lst[j+1]:
                lst[j], lst[j+1] = lst[j+1], lst[j]
                swapped = True

        if not swapped:
            break
```

```
-----> Outer Loop iteration #3

-> Inner Loop iteration #1
Left element: 2
Right element: 1
Not sorted: 2 > 1
Swapping...
Old list: [2, 1, 5, 8, 10]
New list: [1, 2, 5, 8, 10]

-> Inner Loop iteration #2
Left element: 2
Right element: 5
Already sorted: 2 < 5
No change: [1, 2, 5, 8, 10]


-----> Outer Loop iteration #4

-> Inner Loop iteration #1
Left element: 1
Right element: 2
Already sorted: 1 < 2
No change: [1, 2, 5, 8, 10]

There was no need to swap! The list is now sorted
[1, 2, 5, 8, 10]
```

Time to Practice!