

Bubble Sort

Programming and Algorithms

Lecture by
Dr Daniil Osudin

```
n = 3
for i in range(1,n+1):
    print("Hello World!")
```

Hello World!
Hello World!
Hello World!

What will we Cover?

- Bubble sort algorithm
- Understanding the efficiency of the algorithm using big-O notation

Bubble Sort

- In the first pass, each element in the list is compared with the following element in the list.
- If this element is bigger than the element being compared, then they are swapped.
- If you repeat this process enough times, the data will be sorted in ascending order.

Tracing the Behaviour of the Bubble Sort

50	3	15	8	31	26
----	---	----	---	----	----



Swap

Tracing the Behaviour of the Bubble Sort

3	50	15	8	31	26
---	----	----	---	----	----



Swap

Tracing the Behaviour of the Bubble Sort

3	15	50	8	31	26
---	----	----	---	----	----



Swap

Tracing the Behaviour of the Bubble Sort

3	15	8	50	31	26
---	----	---	----	----	----



Swap

Tracing the Behaviour of the Bubble Sort

3	15	8	31	50	26
---	----	---	----	----	----



Swap

Tracing the Behaviour of the Bubble Sort

3	15	8	31	26	50
---	----	---	----	----	----



Do not swap

Tracing the Behaviour of the Bubble Sort

3	15	8	31	26	50
---	----	---	----	----	----



Swap

Tracing the Behaviour of the Bubble Sort

3	8	15	31	26	50
---	---	----	----	----	----



Do not swap

Tracing the Behaviour of the Bubble Sort

3	8	15	31	26	50
---	---	----	----	----	----



Swap

Tracing the Behaviour of the Bubble Sort

3	8	15	26	31	50
---	---	----	----	----	----

Bubble Sort Algorithm

1. `n = len(list)`
2. `for i in range(n-1)`
3. `for j in range(n-1-i)`
4. `if list[j] > list[j + 1]`
5. `temp = list[j]`
6. `list[j] = list[j + 1]`
7. `list[j + 1] = temp`

Bubble Sort Example

```
def bubble_sort(list1):  
    # this function sorts the list using bubble sort algorithm  
    n = len(list1)  
    for i in range(n-1):  
        for j in range(n-1-i):  
            if list1[j] > list1[j+1]:  
                temp = list1[j]  
                list1[j] = list1[j+1]  
                list1[j+1] = temp  
  
input_string = input("Enter your numbers, then press enter: ")  
split_input = input_string.split()  
numbers = [int(n) for n in split_input]  
  
bubble_sort(numbers)  
print("sorted list:", numbers)
```

```
Enter your numbers, then press enter: 8 3 6 1 9 7 0 2 5 4  
sorted list: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```


Analysis

- Makes excessive comparisons.
- Works best on a partially ordered list
- Classical algorithm, good introduction to sorting algorithms.
- Bubble sort uses a nested loop, so in the worst-case scenario, when we have to execute both loops n times, we perform **$O(n^2)$** comparisons
- The actual number of operations is $4n(n + 1)/2 = 2n^2 + 2n = O(n^2)$
- The big-O notation for bubble sort is **$O(n^2)$**

Try It Yourself

Write a program in python environment that takes a string as an input and sorts in alphabetical order using the bubble sort algorithm above