

Bubble Sort

Textbook

Bubble Sort



These next few lessons will cover different sorting algorithms that are useful for sorting data. We will be exploring the following sorting methods:

- Bubble Sorting
- Selection Sorting
- Insertion Sorting
- Merge Sort
- Quicksort

Bubble Sort

Bubble sort is an algorithm that can be used to sort lists into numerical order. Bubble sort works by comparing two adjacent numbers, and, if necessary, swapping them into the correct order.

Bubble sorting only works with number information. It does not work with strings.

At the end of each iteration the largest number is added to the top of the list. As the algorithm repeatedly loops, the list of numbers is slowly sorted from smallest to largest.

6 5 3 1 8 7 2 4

Source: Swfung8 / CC BY-SA (<https://creativecommons.org/licenses/by-sa/3.0>)

Pseudocode

Pseudocode is a way to write out the concept of a program that isn't language specific. It looks like computer code, but it's written in a more readable way to help readers understand it better. Each of these sorting algorithm lessons will have the algorithm written out in pseudocode to help you understand it better.

Here is pseudocode for bubble sort:

```
1 function BubbleSort(list)
2
3   for all elements in list:
4     if list[i] > list[i+1]
5       swap(list[i], list[i+1])
6     end if
7   end for
8   return list
9 end BubbleSort
```

Copy

Bubble Sort is very inefficient, and as we move to better sorting algorithms you'll notice it gets much faster. Programs that sort very large data sets using bubble sorts take a long time and use a lot of energy to process.

Checkpoint

Bubble Sort

1. Create a program that will bubble sort a list in Python.

Consider the following list:

```
arr = [10, 32, 4, 90, 15, 20, 89, 1, 3, 45, 42, 87, 91, 18, 25, 76, 38, 12]
```

2. The program will then take in a number input from the user.
3. The program will add the number to the list.
4. The program will sort the list using bubble sort.
5. Print the resulting list.

For example:

Input: 22

Output: [1, 3, 4, 10, 12, 15, 18, 20, 22, 25, 32, 38, 42, 45, 76, 87, 89, 90, 91]

Another example:

Input: 99

Output: [1, 3, 4, 10, 12, 15, 18, 20, 25, 32, 38, 42, 45, 76, 87, 89, 90, 91, 99]

Hint: To swap values, use the following code for ideas:

```
sort_list[j], sort_list[j+1] = sort_list[j+1], sort_list[j]
```

Questions (3)

1. True or False: Bubble sorting works well with integer or string data types.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

2. True or False: Bubble sorting is highly efficient.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

3. In bubble sorting, what happens at the end of each iteration?

MULTIPLE CHOICE

Choose the correct answer:

- A. The largest number ends up at the top of the list.
- B. The smallest number ends up at the bottom of the list.
- C. The center number ends up in the middle of the list.
- D. The list gets reversed.

Answer Keys & Solutions

Checkpoint Solutions

Bubble Sort

```
1 answer = int(input("Enter a number 0 - 100"))
2
3 arr = [10, 32, 4, 90, 15, 20, 89, 1, 3, 45, 42, 87, 91, 18, 25, 76, 38, 12]
4
5 arr.append(answer)
6
7 def bubbleSort(sort_list):
8     n = len(sort_list)
9     for i in range(n-1):
10         for j in range(0, n-i-1):
11             if sort_list[j] > sort_list[j+1] :
12                 sort_list[j], sort_list[j+1] = sort_list[j+1], sort_list[j]
13         return sort_list
14
15 bubbleSort(arr)
16
17 print(arr)
```

Questions

1. True or False: Bubble sorting works well with integer or string data types.

MULTIPLE CHOICE

Correct Answer:

A. True

✗ Incorrect

B. False

✓ Correct

Explanation:

Bubble sorting helps put integers into numerical order--this doesn't work with strings.

2. True or False: Bubble sorting is highly efficient.

MULTIPLE CHOICE

Correct Answer:

A. True

✗ Incorrect

B. False

✓ Correct

Explanation:

Bubble sorting is highly inefficient. For large data sets, it takes a long time and takes a lot of computational power.

3. In bubble sorting, what happens at the end of each iteration?

MULTIPLE CHOICE

Correct Answer:

A. The largest number ends up at the top of the list.

✓ Correct

B. The smallest number ends up at the bottom of the list.

✗ Incorrect

C. The center number ends up in the middle of the list.

✗ Incorrect

D. The list gets reversed.

✗ Incorrect

Explanation:

The two adjacent numbers are compared for size and the largest one moves to the right through the whole list.