

Merge Sort

Textbook

Merge Sort



Overview

The Merge sort algorithm involves dividing the list into n sublists, where each sublist contains one element of the main list. Each of these lists is considered to be sorted because they're only one element in size.

Once the sublists have been created, merge the sublists to produce new sorted sublists until only one list remains. This list will be the sorted list.

Merge sort is a recursive algorithm, meaning that the Merge sort algorithm calls itself when it needs to create a sublist. You'll see how this works in the pseudocode section.

6 5 3 1 8 7 2 4

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

Example

The algorithm for merge sort needs two functions – a function that divides the list into sublists, and a function that merges them back together in order.

```
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 mergeSort(sort_list):
8     if len(sort_list) > 1:
9
10         mid = len(sort_list)//2
11         L = sort_list[:mid]
12         R = sort_list[mid:]
13         mergeSort(L)
14         mergeSort(R)
15
16         i = j = k = 0
17
18         while i < len(L) and j < len(R):
19             if L[i] < R[j]:
20                 sort_list[k] = L[i]
21                 i += 1
22             else:
23                 sort_list[k] = R[j]
24                 j += 1
25             k += 1
26
27         while i < len(L):
28             sort_list[k] = L[i]
29             i += 1
30             k += 1
31
```

```

32         while j < len(R):
33             sort_list[k] = R[j]
34             j += 1
35             k += 1
36
37 mergeSort(arr)
38 print(arr)

```

Merge sorting is extremely fast because it is a recursive algorithm. Recursion helps programs to run quickly.

Checkpoint

Merge Sort

1. Create a program that will merge 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 merge sort.
5. Print the resulting list.

For example:

Input: 2

Output: [1, 2, 3, 4, 10, 12, 15, 18, 20, 25, 32, 38, 42, 45, 76, 87, 89, 90, 91]

Another example:

Input: 49

Output: [1, 3, 4, 10, 12, 15, 18, 20, 25, 32, 38, 42, 45, 49, 76, 87, 89, 90, 91]

Hint: This function will *call itself* inside the function. This is recursion.

Questions (3)

1. What is it called when the function calls itself? Merge sorting uses this technique.

MULTIPLE CHOICE

Choose the correct answer:

- A. recursion
- B. looping
- C. iteration
- D. abstraction

2. Of the kinds of sorting we have learned about so far, which is the fastest?**Choose the correct answer:**

- A. Bubble sorting
- B. Merge sorting
- C. Insertion sorting
- D. Selection sorting

3. The merge sort algorithm divides a list into what?**Choose the correct answer:**

- A. sublists, usually halves
- B. pairs
- C. randomly-sized arrays
- D. positive and negative numbers

Answer Keys & Solutions

Checkpoint Solutions

Merge 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 mergeSort(sort_list):
8     if len(sort_list) > 1:
9
10         mid = len(sort_list)//2
11         L = sort_list[:mid]
12         R = sort_list[mid:]
13         mergeSort(L)
14         mergeSort(R)
15
16         i = j = k = 0
17
18         while i < len(L) and j < len(R):
19             if L[i] < R[j]:
20                 sort_list[k] = L[i]
21                 i += 1
22             else:
23                 sort_list[k] = R[j]
24                 j += 1
25             k += 1
26
27         while i < len(L):
28             sort_list[k] = L[i]
29             i += 1
30             k += 1
31
32         while j < len(R):
33             sort_list[k] = R[j]
34             j += 1
35             k += 1
36
37 mergeSort(arr)
38 print(arr)
```

Questions

1. What is it called when the function calls itself? Merge sorting uses this technique.

MULTIPLE CHOICE

Correct Answer:

- | | |
|----------------|-------------|
| A. recursion | ✓ Correct |
| B. looping | ✗ Incorrect |
| C. iteration | ✗ Incorrect |
| D. abstraction | ✗ Incorrect |

Explanation:

Iteration is where you get feedback on code and tweak it to try again. Abstraction is simply representing complex info

2. Of the kinds of sorting we have learned about so far, which is the fastest?

MULTIPLE CHOICE

Correct Answer:

- | | |
|----------------------|-------------|
| A. Bubble sorting | ✗ Incorrect |
| B. Merge sorting | ✓ Correct |
| C. Insertion sorting | ✗ Incorrect |
| D. Selection sorting | ✗ Incorrect |

Explanation:

Merge sorting uses recursion--which makes it really fast.

3. The merge sort algorithm divides a list into what?

MULTIPLE CHOICE

Correct Answer:

- | | |
|----------------------------------|-------------|
| A. sublists, usually halves | ✓ Correct |
| B. pairs | ✗ Incorrect |
| C. randomly-sized arrays | ✗ Incorrect |
| D. positive and negative numbers | ✗ Incorrect |

Explanation:

Each list gets broken down into separate lists.