**EX NO: 7 MERGE SORT**

**Date :**

**AIM**

To write a python program to sort the given list using merge sort algorithm.

Input:[5,8,2]

Output:[2,5,8]

Input:[6,2,8,4,3,7,5,1]

Output:[1,2,3,4,5,6,7,8]

Input=[6,-2,8,4,-3,7,-5,1]

Output=[-5,-3,-2,1,4,6,7,8]

Input=[6,-2,"8.8",4.4,-3,7.5,-5,1]

Output="Invalid input"

Input=23

Output="Invalid input"

**ALGORITHM**

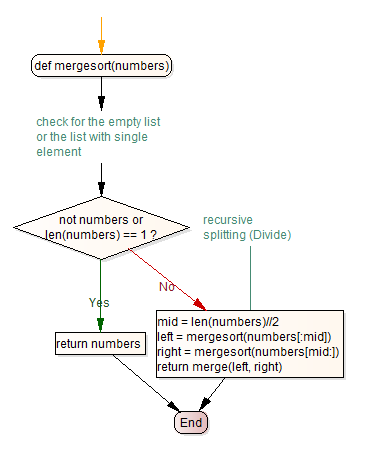
**Step1:** Check for the empty list or the list with single element

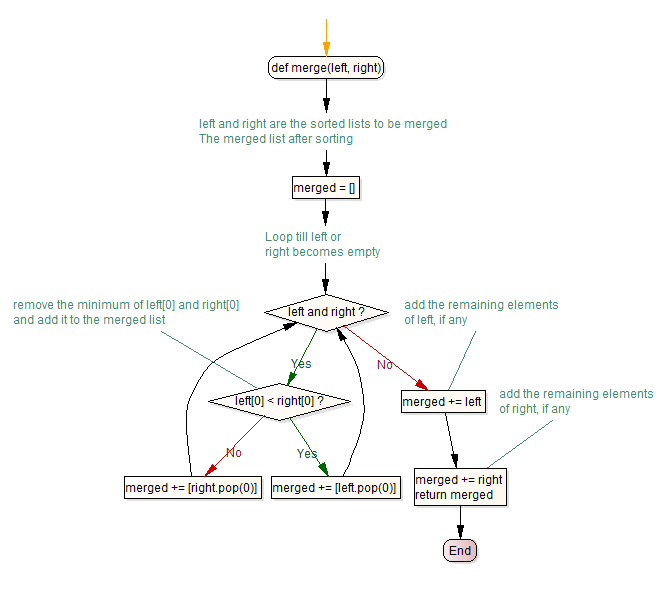
**Step2:**  If not numbers or len(numbers) == 1, apply recursive splitting

**Step 3:** Remove the minimum of left[0] and right[0] and add it to the merged list

**Step 4:** Repeat until list is sorted

**FLOWCHART**



****

**PRE-LAB QUESTIONS**

1. How do you get the number of elements in the list
2. Merge two lists [2,4,3] and [7,5,6]
3. Consider left and right lists of size 1. Merge them in a sorted order.

Example:

left = [12] right = [3]

merged = [3,12]

1. a. Now consider the two sorted lists of unspecified size. Merge them in a sorted order.

Example:

left = [12,45] right = [3,17]

merged = [3,12,17,45]

b. Merge two sorted lists [2,6,7] and [3,5,8] in the sorted order

1. Divide the list num into left and right halves.

Example:

num = [6,2,8,4,3,7,5,1]

left = [6,2,8,4] right = [3,7,5,1]

1. Recursively divide, till the partition size is 1

Example:

num = [12,3,45,17,15]

left = [12,3]

left = [12]

right = [3]

right = [17, 15]

left = [17]

right = [15]

1. What are the first two elements to be merged from the input list [6,2,8,4,3,7,5,1] in the mergesort
2. Validate whether given input is of list type.
3. Validate whether all the elements in the input are of int type.
4. Validate whether given list is empty

**SOURCE CODE**

def mergesort(numbers):

# check for the empty list or the list with single element

if not numbers or len(numbers) == 1:

return numbers

# recursive splitting (Divide)

else:

mid = len(numbers)//2

left = mergesort(numbers[:mid])

right = mergesort(numbers[mid:])

return merge(left, right)

def merge(left, right):

# left and right are the sorted lists to be merged

# The merged list after sorting

merged = []

# Loop till left or right becomes empty

while left and right:

# remove the minimum of left[0] and right[0]

# and add it to the merged list

if left[0] < right[0]:

merged += [left.pop(0)]

else:

merged += [right.pop(0)]

# add the remaining elements of left, if any

merged += left

# add the remaining elements of right, if any

merged += right

return merged

**OUTPUT**

Enter number: 6,2,8,4,3,7,5,1

Output : 1,2,3,4,5,6,7,8

**RESULT**

Thus the python program for performing the merge sort was executed and verified successfully.

**POST-LAB QUESTIONS**

1. Modify the program to validate whether given input is a valid list.
2. Modify the code to sort the list of string elements.

Example:

input=['python','c','c++','java']

expected output=['c','c++','python','java']

**REFERENCE**

1. Animation - <https://visualgo.net/en/sorting?slide=10> slides - <http://bit.ly/KG_mergesort>