LAPORAN TUGAS
PRAKTIKUM ALGORITMA & STRUKTUR DATA
MODUL 6

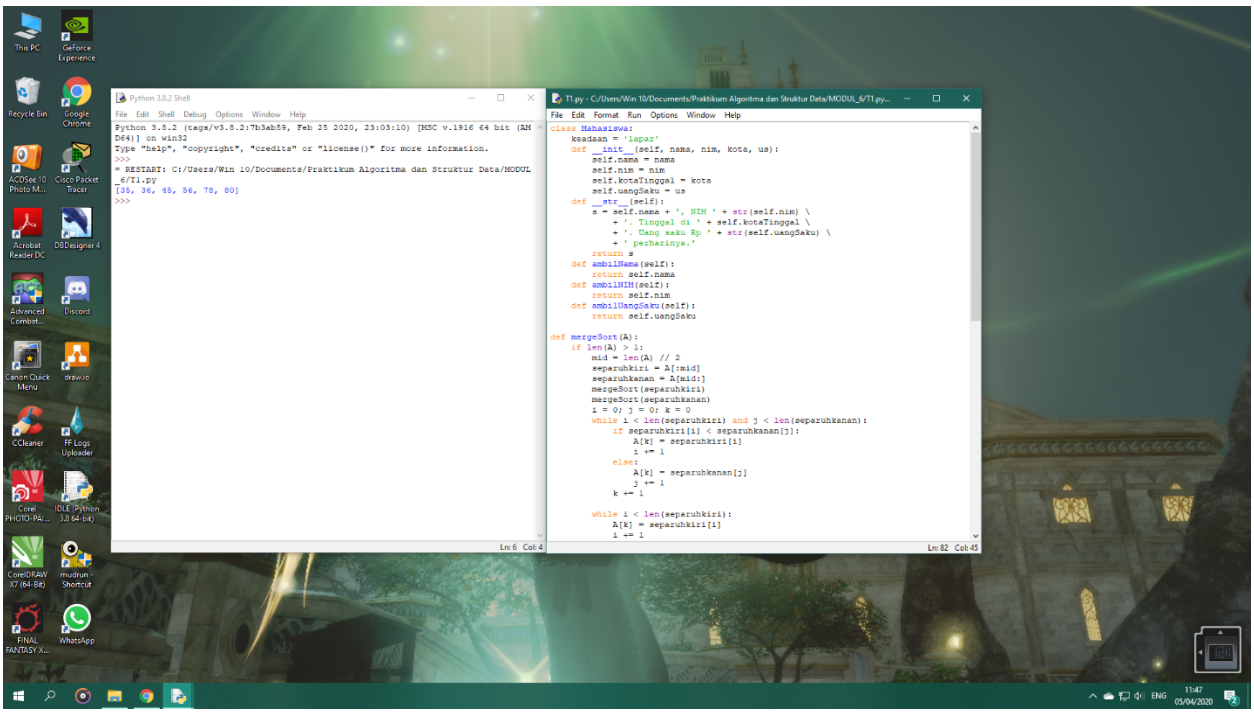
Nama : Muhammad Ridwan NurFarizi

NIM : L200180020

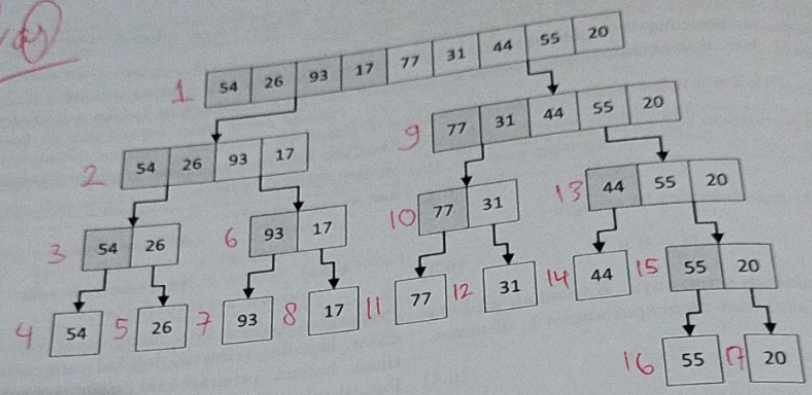
Kelas : A

Tugas

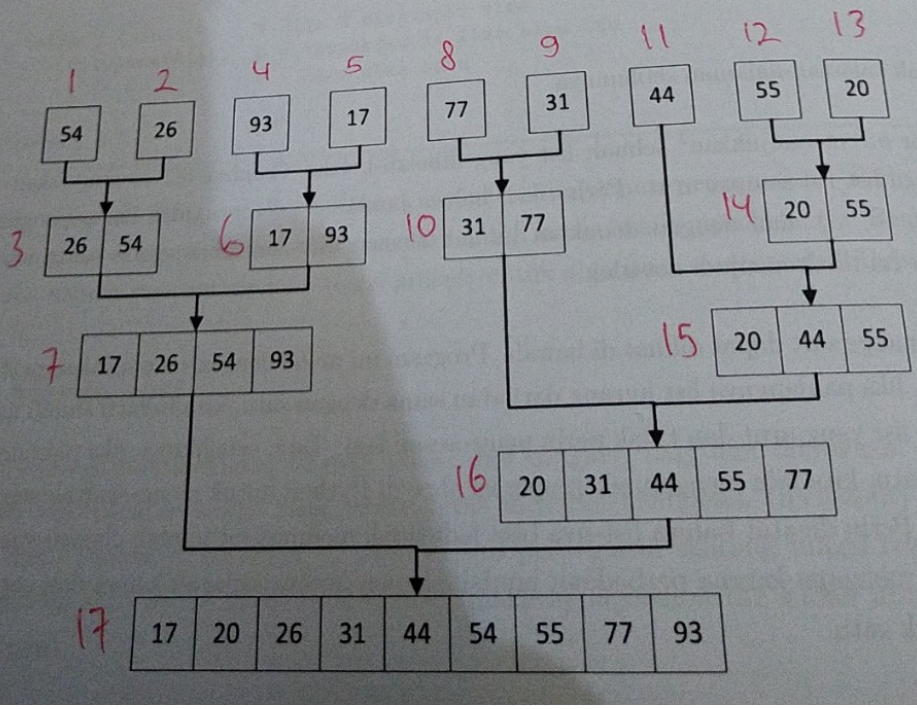
No. 1



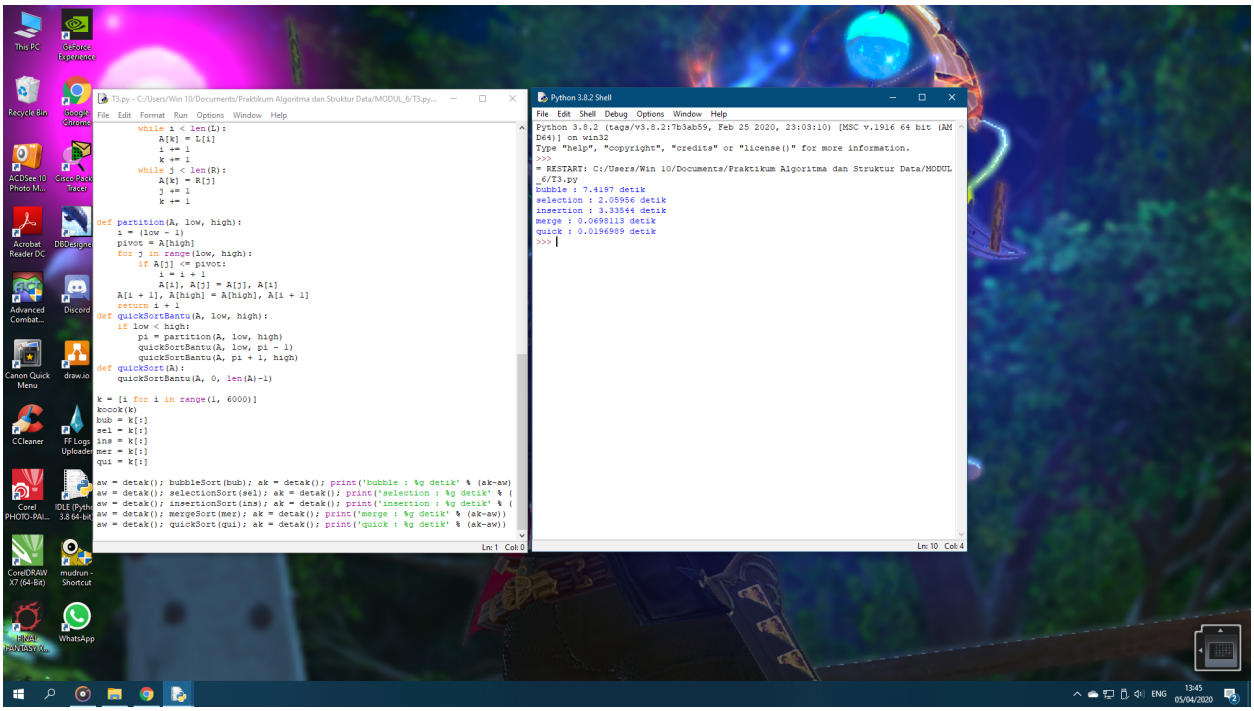
No. 2



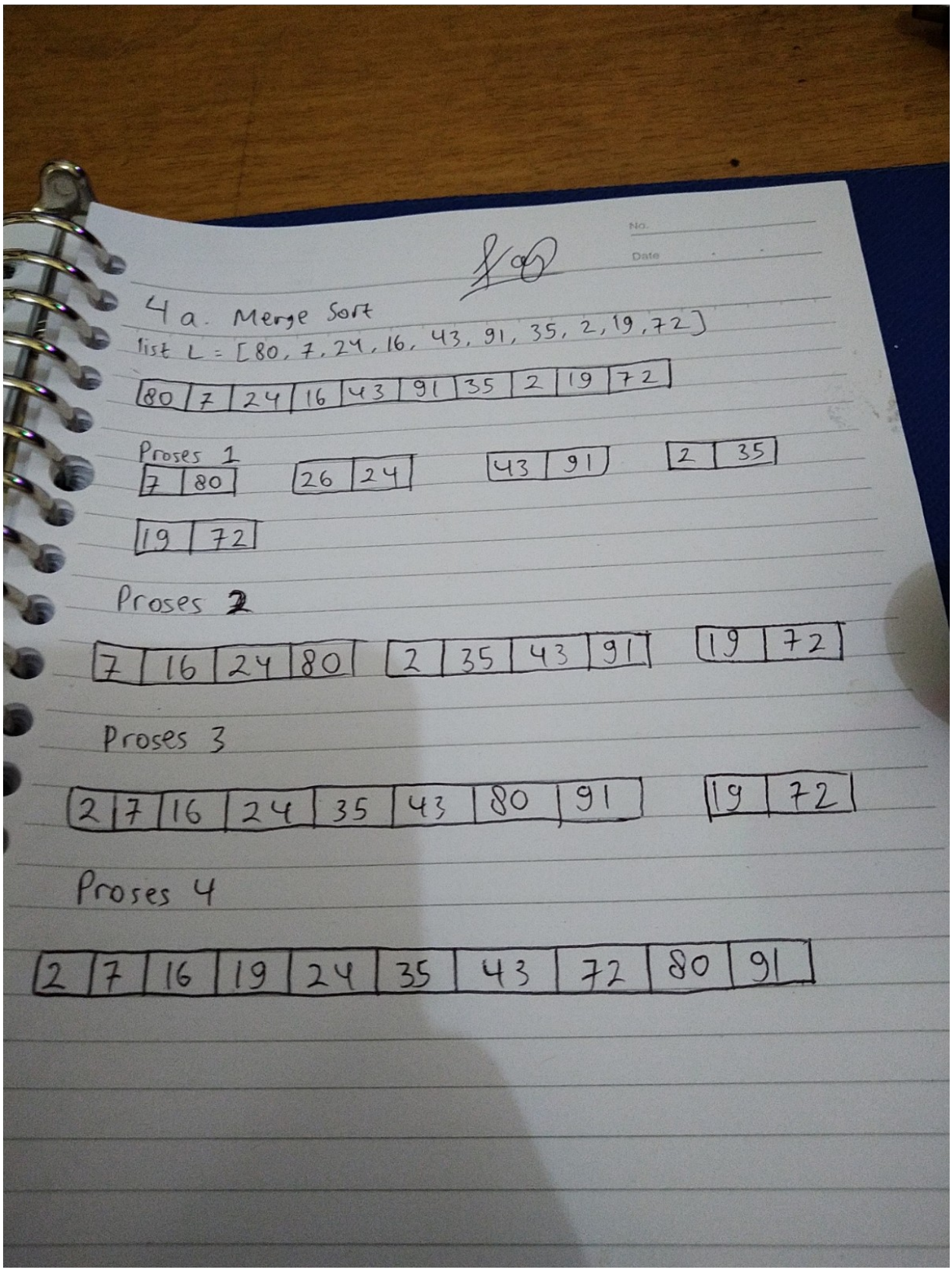
Gambar 6.1: Membelah list sampai tiap sub-list berisi satu elemen atau kosong. Sesudah itu digabung seperti ditunjukkan di Gambar 6.2.



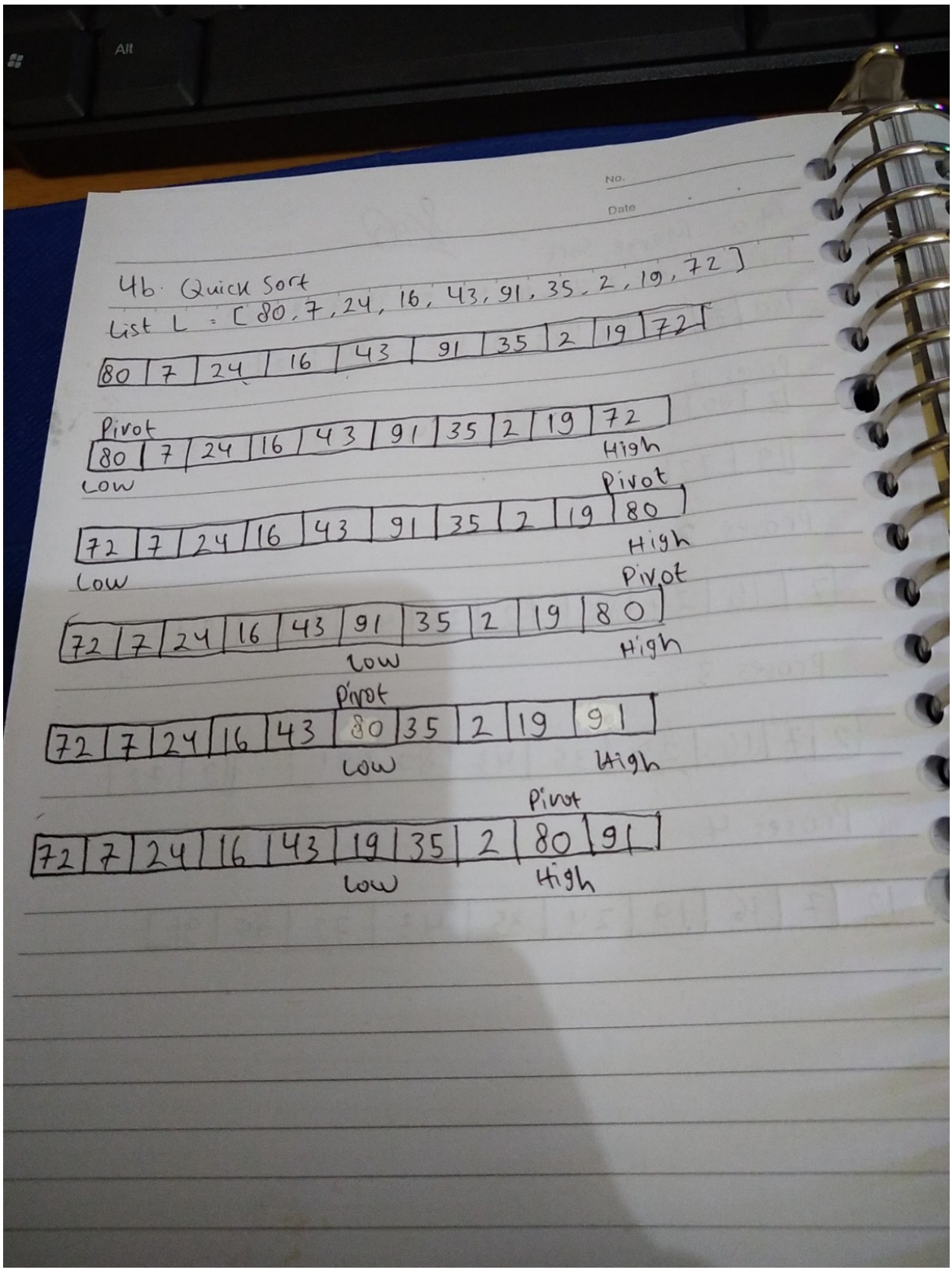
No. 3


T3.py - C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL_6/T3.py...
File Edit Format Run Options Window Help
while i < len(L):
A[k] = L[i]
i += 1
k += 1
while j < len(R):
A[k] = R[j]
j += 1
k += 1
def partition(A, low, high):
i = (low - 1)
pivot = A[high]
for j in range(low, high):
if A[j] <= pivot:
i = i + 1
A[i], A[j] = A[j], A[i]
A[i + 1], A[high] = A[high], A[i + 1]
return i + 1
def quickSortBantu(A, low, high):
if low < high:
pi = partition(A, low, high)
quickSortBantu(A, low, pi - 1)
quickSortBantu(A, pi + 1, high)
def quickSort(A):
quickSortBantu(A, 0, len(A)-1)
k = [i for i in range(1, 6000)]
kocok(k)
bub = k[:]
sel = k[:]
ins = k[:]
mer = k[:]
qui = k[:]
aw = detak(); bubbleSort(bub); ak = detak(); print('bubble : %g detik' % (ak-aw)
aw = detak(); selectionSort(sel); ak = detak(); print('selection : %g detik' % (
aw = detak(); insertionSort(ins); ak = detak(); print('insertion : %g detik' % (
aw = detak(); mergeSort(mer); ak = detak(); print('merge : %g detik' % (ak-aw))
aw = detak(); quickSort(qui); ak = detak(); print('quick : %g detik' % (ak-aw))
Ln: 1 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL
_6/T3.py
bubble : 7.4197 detik
selection : 2.05956 detik
insertion : 3.33544 detik
merge : 0.0698113 detik
quick : 0.0196989 detik
>>>
Ln: 10 Col: 4
This PC
GeForce Experience
Recycle Bin
Google Chrome
ACDSee 10 Photo M...
Cisco Packet Tracer
Acrobat Reader DC
DBDesigner
Advanced Combat...
Discord
Canon Quick Menu
draw.io
CCleaner
FF Logs Uploader
Corel PHOTO-PAI...
IDLE (Pyth... 3.8 64-bit
CorelDRAW X7 (64-Bit)
mudrun - Shortcut
FINAL FANTASY X...
WhatsApp
ENG
13:45
05/04/2020

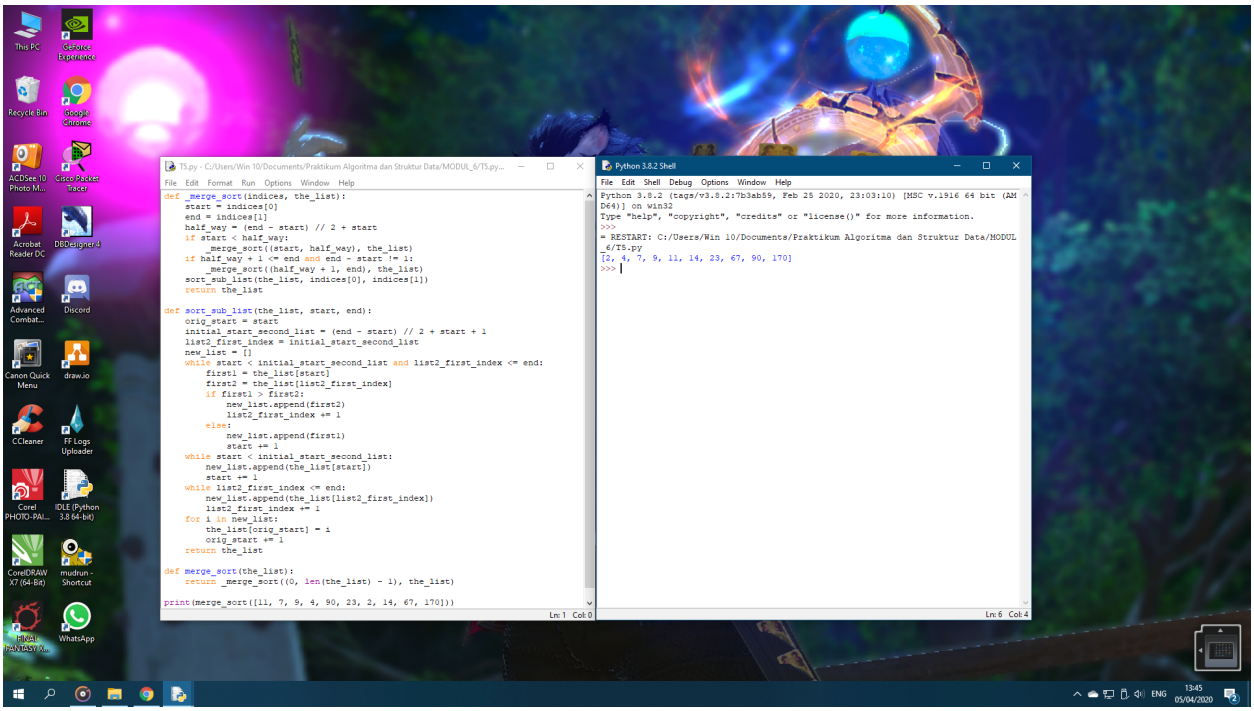
No. 4A


No.
Date
4a. Merge Sort
list L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72]
80 7 24 16 43 91 35 2 19 72
Proses 1
7 80
26 24
43 91
2 35
19 72
Proses 2
7 16 24 80
2 35 43 91
19 72
Proses 3
2 7 16 24 35 43 80 91
19 72
Proses 4
2 7 16 19 24 35 43 72 80 91

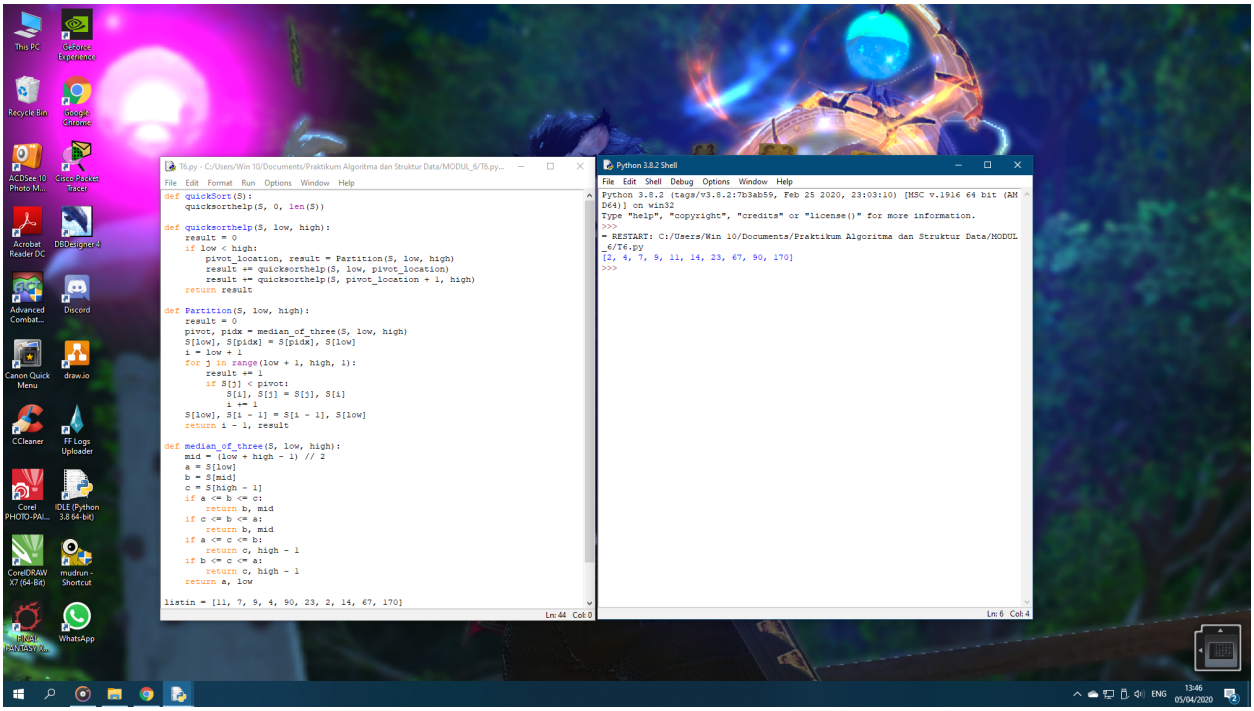
No. 4B


4b. Quick Sort
List L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72]
80 7 24 16 43 91 35 2 19 72
Pivot
80 7 24 16 43 91 35 2 19 72
Low
High
72 7 24 16 43 91 35 2 19 80
Pivot
Low
High
72 7 24 16 43 91 35 2 19 80
Pivot
Low
High
72 7 24 16 43 80 35 2 19 91
Pivot
Low
High
72 7 24 16 43 19 35 2 80 91
Low
High

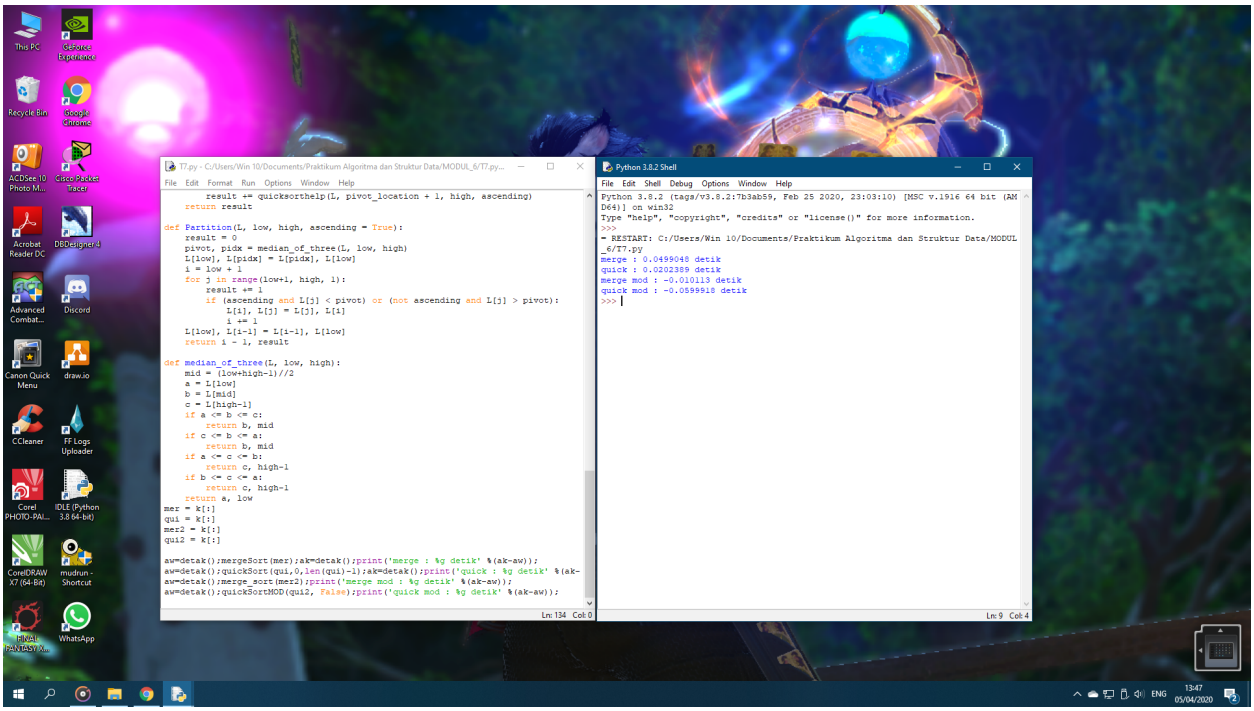
No. 5


T5.py - C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL_6/T5.py...
File Edit Format Run Options Window Help
def _merge_sort(indices, the_list):
start = indices[0]
end = indices[1]
half_way = (end - start) // 2 + start
if start < half_way:
_merge_sort((start, half_way), the_list)
if half_way + 1 <= end and end - start != 1:
_merge_sort((half_way + 1, end), the_list)
sort_sub_list(the_list, indices[0], indices[1])
return the_list
def sort_sub_list(the_list, start, end):
orig_start = start
initial_start_second_list = (end - start) // 2 + start + 1
list2_first_index = initial_start_second_list
new_list = []
while start < initial_start_second_list and list2_first_index <= end:
first1 = the_list[start]
first2 = the_list[list2_first_index]
if first1 > first2:
new_list.append(first2)
list2_first_index += 1
else:
new_list.append(first1)
start += 1
while start < initial_start_second_list:
new_list.append(the_list[start])
start += 1
while list2_first_index <= end:
new_list.append(the_list[list2_first_index])
list2_first_index += 1
for i in new_list:
the_list[orig_start] = i
orig_start += 1
return the_list
def merge_sort(the_list):
return _merge_sort((0, len(the_list) - 1), the_list)
print(merge_sort([11, 7, 9, 4, 90, 23, 2, 14, 67, 170]))
Ln: 1 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL_6/T5.py
[2, 4, 7, 9, 11, 14, 23, 67, 90, 170]
>>>
Ln: 6 Col: 4

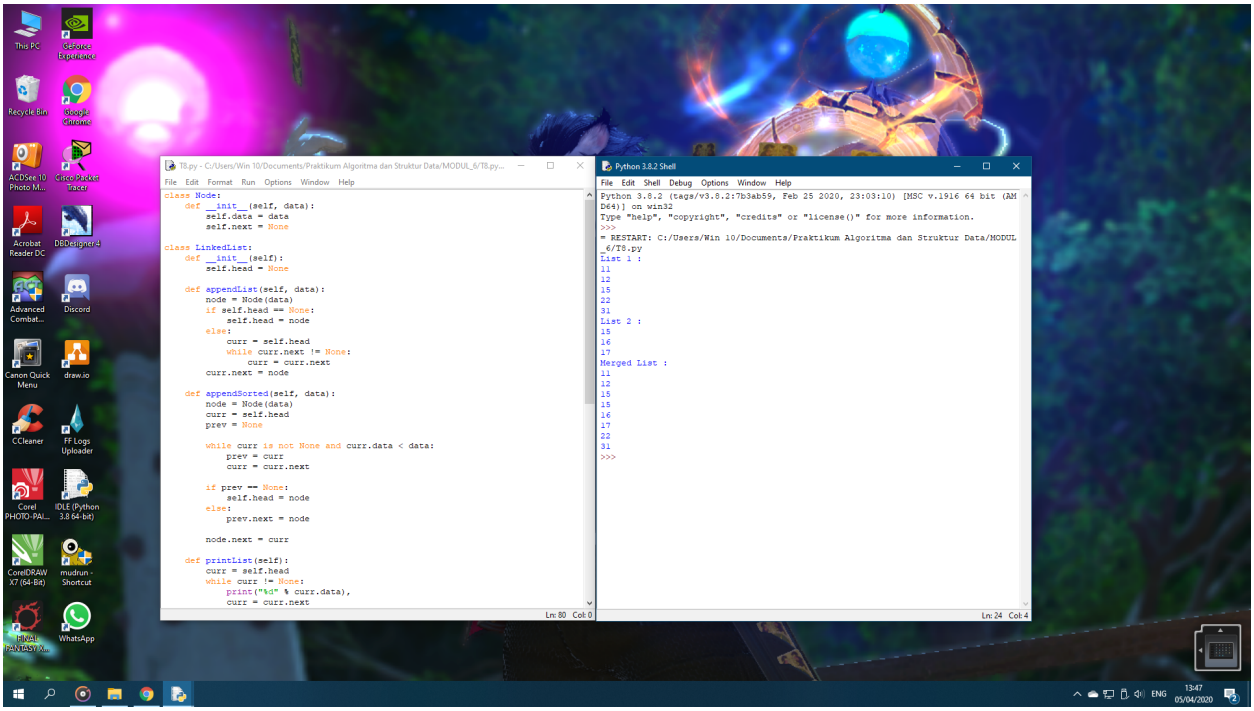
No. 6


T6.py - C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL_6/T6.py...
File Edit Format Run Options Window Help
def quickSort(S):
quicksorthelp(S, 0, len(S))
def quicksorthelp(S, low, high):
result = 0
if low < high:
pivot_location, result = Partition(S, low, high)
result += quicksorthelp(S, low, pivot_location)
result += quicksorthelp(S, pivot_location + 1, high)
return result
def Partition(S, low, high):
result = 0
pivot, pidx = median_of_three(S, low, high)
S[low], S[pidx] = S[pidx], S[low]
i = low + 1
for j in range(low + 1, high, 1):
result += 1
if S[j] < pivot:
S[i], S[j] = S[j], S[i]
i += 1
S[low], S[i - 1] = S[i - 1], S[low]
return i - 1, result
def median_of_three(S, low, high):
mid = (low + high - 1) // 2
a = S[low]
b = S[mid]
c = S[high - 1]
if a <= b <= c:
return b, mid
if c <= b <= a:
return b, mid
if a <= c <= b:
return c, high - 1
if b <= c <= a:
return c, high - 1
return a, low
listin = [11, 7, 9, 4, 90, 23, 2, 14, 67, 170]
Ln: 44 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL_6/T6.py
[2, 4, 7, 9, 11, 14, 23, 67, 90, 170]
>>>
Ln: 6 Col: 4

No. 7


T7.py - C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL_6/T7.py...
File Edit Format Run Options Window Help
result += quicksorthelp(L, pivot_location + 1, high, ascending)
return result
def Partition(L, low, high, ascending = True):
result = 0
pivot, pidx = median_of_three(L, low, high)
L[low], L[pidx] = L[pidx], L[low]
i = low + 1
for j in range(low+1, high, 1):
result += 1
if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot):
L[i], L[j] = L[j], L[i]
i += 1
L[low], L[i-1] = L[i-1], L[low]
return i - 1, result
def median_of_three(L, low, high):
mid = (low+high-1)//2
a = L[low]
b = L[mid]
c = L[high-1]
if a <= b <= c:
return b, mid
if c <= b <= a:
return b, mid
if a <= c <= b:
return c, high-1
if b <= c <= a:
return c, high-1
return a, low
mer = k[:]
qui = k[:]
mer2 = k[:]
qui2 = k[:]
aw=detak();mergeSort(mer);ak=detak();print('merge : %g detik' %(ak-aw));
aw=detak();quickSort(qui,0,len(qui)-1);ak=detak();print('quick : %g detik' %(ak-
aw=detak();merge_sort(mer2);print('merge mod : %g detik' %(ak-aw));
aw=detak();quickSortMOD(qui2, False);print('quick mod : %g detik' %(ak-aw));
Ln: 134 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL
_6/T7.py
merge : 0.0499048 detik
quick : 0.0202389 detik
merge mod : -0.010113 detik
quick mod : -0.0599918 detik
>>>
Ln: 9 Col: 4

No. 8


T8.py - C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL_6/T8.py...
File Edit Format Run Options Window Help
class Node:
def __init__(self, data):
self.data = data
self.next = None
class LinkedList:
def __init__(self):
self.head = None
def appendList(self, data):
node = Node(data)
if self.head == None:
self.head = node
else:
curr = self.head
while curr.next != None:
curr = curr.next
curr.next = node
def appendSorted(self, data):
node = Node(data)
curr = self.head
prev = None
while curr is not None and curr.data < data:
prev = curr
curr = curr.next
if prev == None:
self.head = node
else:
prev.next = node
node.next = curr
def printList(self):
curr = self.head
while curr != None:
print("%d" % curr.data),
curr = curr.next
Ln: 80 Col: 0
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Win 10/Documents/Praktikum Algoritma dan Struktur Data/MODUL
_6/T8.py
List 1 :
11
12
15
22
31
List 2 :
15
16
17
Merged List :
11
12
15
15
16
17
22
31
>>>
Ln: 24 Col: 4