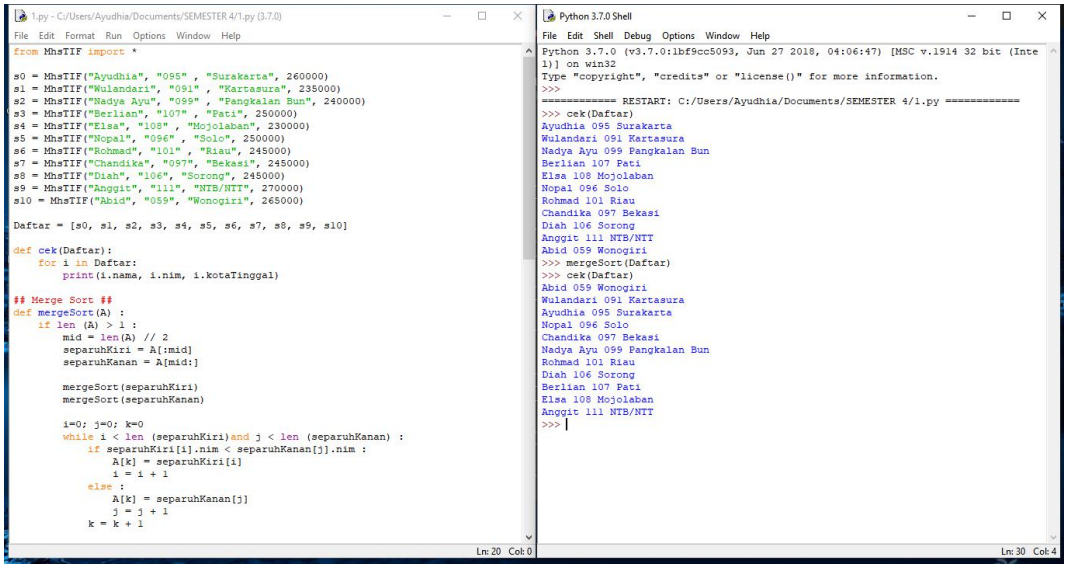
Nama : Ayudhia Isnafiani Fanada
NIM : L200180095
Kelas : D

ALGORITMA DAN STRUKTUR DATA
MODUL 6

6.4 Soal-soal untuk Mahasiswa

1. Ubahlah kode mergeSort dan quickSort di atas agar bisa mengurutkan list yang berisi object-object mhsTIF yang sudah kamu buat di modul 2.

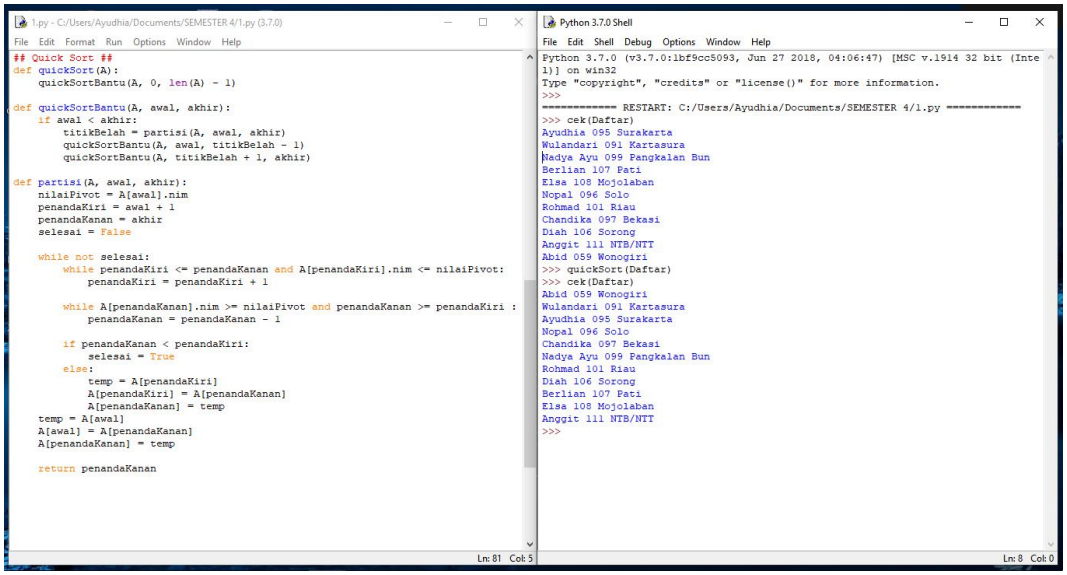
a) Merge Sort



```
from MhsTIF import *

s0 = MhsTIF("Ayudhia", "095" , "Surakarta", 260000)
s1 = MhsTIF("Wulandari", "091" , "Kartasura", 235000)
s2 = MhsTIF("Nadya Ayu", "099" , "Pangkalan Bun", 240000)
s3 = MhsTIF("Berlian", "107" , "Pati", 250000)
s4 = MhsTIF("Elsa", "108" , "Mojolaban", 230000)
s5 = MhsTIF("Nopal", "096" , "Solo", 250000)
s6 = MhsTIF("Rohmad", "101" , "Riau", 245000)
s7 = MhsTIF("Chandika", "097", "Bekasi", 245000)
s8 = MhsTIF("Diah", "106", "Sorong", 245000)
s9 = MhsTIF("Anggit", "111", "NTB/NTT", 270000)
s10 = MhsTIF("Abid", "059", "Wonogiri", 265000)

Daftar = [s0, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10]

def cek(Daftar):
    for i in Daftar:
        print(i.nama, i.nim, i.kotaTinggal)

## Merge Sort ##
def mergeSort(A) :
    if len (A) > 1 :
        mid = len(A) // 2
        separuhKiri = A[:mid]
        separuhKanan = A[mid:]

        mergeSort(separuhKiri)
        mergeSort(separuhKanan)

        i=0; j=0; k=0
        while i < len (separuhKiri)and j < len (separuhKanan) :
            if separuhKiri[i].nim < separuhKanan[j].nim :
                A[k] = separuhKiri[i]
                i = i + 1
            else :
                A[k] = separuhKanan[j]
                j = j + 1
            k = k + 1
```

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Ayudhia/Documents/SEMESTER 4/1.py ==========
>>> cek(Daftar)
Ayudhia 095 Surakarta
Wulandari 091 Kartasura
Nadya Ayu 099 Pangkalan Bun
Berlian 107 Pati
Elsa 108 Mojolaban
Nopal 096 Solo
Rohmad 101 Riau
Chandika 097 Bekasi
Diah 106 Sorong
Anggit 111 NTB/NTT
Abid 059 Wonogiri
>>> mergeSort(Daftar)
>>> cek(Daftar)
Abid 059 Wonogiri
Wulandari 091 Kartasura
Ayudhia 095 Surakarta
Nopal 096 Solo
Chandika 097 Bekasi
Nadya Ayu 099 Pangkalan Bun
Rohmad 101 Riau
Diah 106 Sorong
Berlian 107 Pati
Elsa 108 Mojolaban
Anggit 111 NTB/NTT
>>>
```

b) Quick Sort

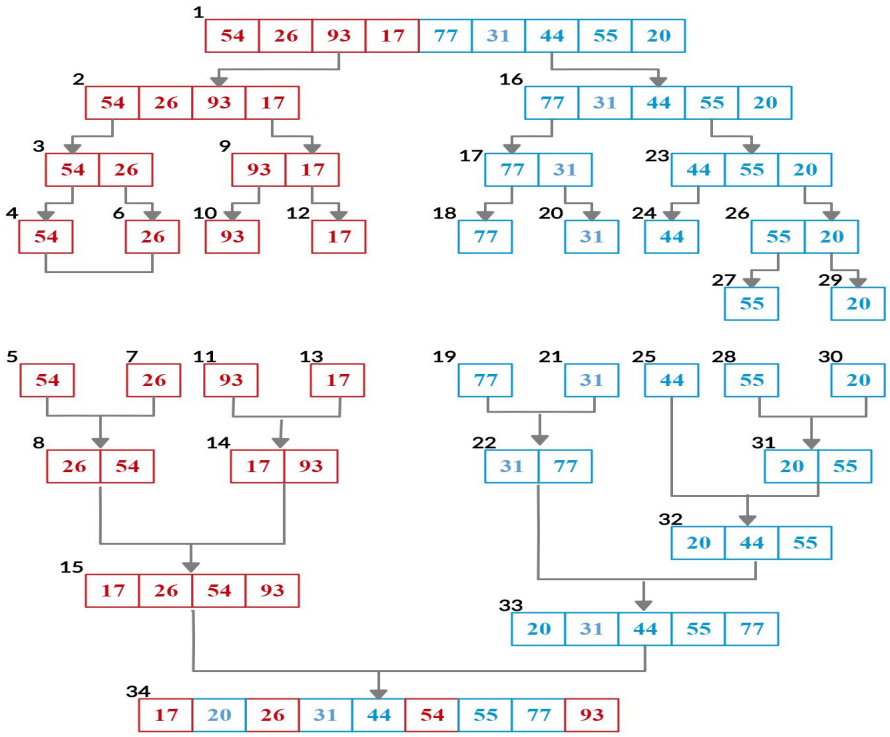


```
## Quick Sort ##
def quickSort(A):
    quickSortBantu(A, 0, len(A) - 1)

def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah = partisi(A, awal, akhir)
        quickSortBantu(A, awal, titikBelah - 1)
        quickSortBantu(A, titikBelah + 1, akhir)

def partisi(A, awal, akhir):
    nilaiPivot = A[awal].nim
    penandaKiri = awal + 1
    penandaKanan = akhir
    selesai = False

    while not selesai:
        while penandaKiri <= penandaKanan and A[penandaKiri].nim <= nilaiPivot:
            penandaKiri = penandaKiri + 1

        while A[penandaKanan].nim >= nilaiPivot and penandaKanan >= penandaKiri :
            penandaKanan = penandaKanan - 1

        if penandaKanan < penandaKiri:
            selesai = True
        else:
            temp = A[penandaKiri]
            A[penandaKiri] = A[penandaKanan]
            A[penandaKanan] = temp
    temp = A[awal]
    A[awal] = A[penandaKanan]
    A[penandaKanan] = temp

    return penandaKanan
```

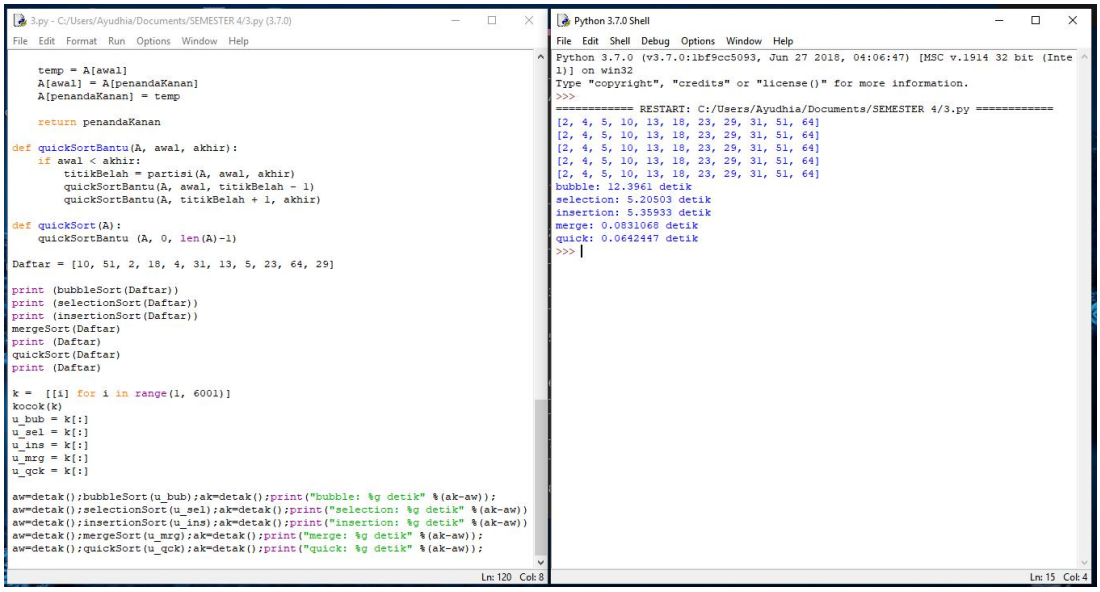
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Ayudhia/Documents/SEMESTER 4/1.py ==========
>>> cek(Daftar)
Ayudhia 095 Surakarta
Wulandari 091 Kartasura
Nadya Ayu 099 Pangkalan Bun
Berlian 107 Pati
Elsa 108 Mojolaban
Nopal 096 Solo
Rohmad 101 Riau
Chandika 097 Bekasi
Diah 106 Sorong
Anggit 111 NTB/NTT
Abid 059 Wonogiri
>>> quickSort(Daftar)
>>> cek(Daftar)
Abid 059 Wonogiri
Wulandari 091 Kartasura
Ayudhia 095 Surakarta
Nopal 096 Solo
Chandika 097 Bekasi
Nadya Ayu 099 Pangkalan Bun
Rohmad 101 Riau
Diah 106 Sorong
Berlian 107 Pati
Elsa 108 Mojolaban
Anggit 111 NTB/NTT
>>>
```

2. Memakai bolpen merah atau biru, tandai dan beri nomer urut eksekusi proses pada Gambar 6.1 dan 6.2, dengan mengacu pada output halaman 59.

GAMBAR 6.1 DAN 6.2



3. Uji kecepatan. Ujilah mergeSort dan quickSort diatas (bersama metode sort yang kamu punya sebelumnya).

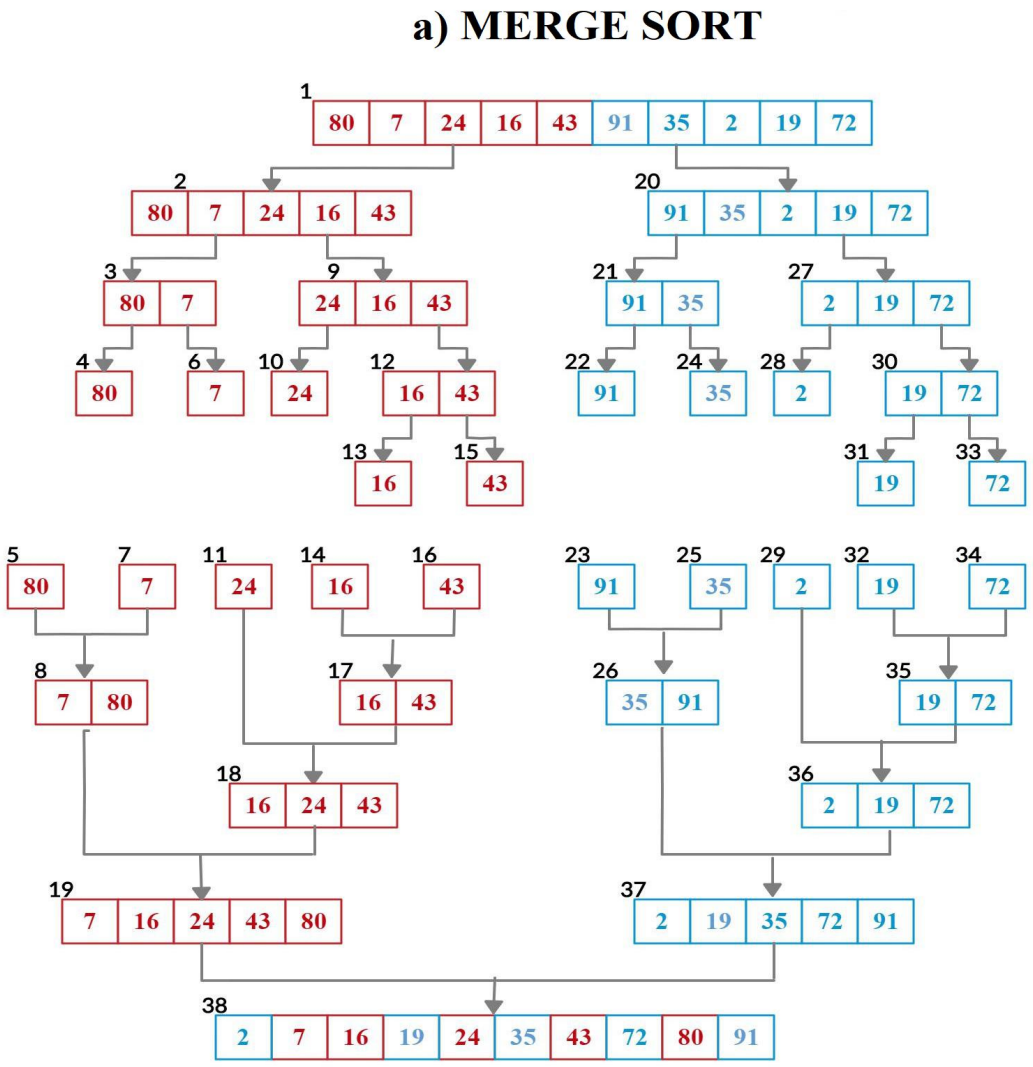


```
    temp = A[awal]
    A[awal] = A[penandaKanan]
    A[penandaKanan] = temp

    return penandaKanan

def quickSortBantu(A, awal, akhir):
    if awal < akhir:
        titikBelah = partisi(A, awal, akhir)
        quickSortBantu(A, awal, titikBelah - 1)
        quickSortBantu(A, titikBelah + 1, akhir)

def quickSort(A):
    quickSortBantu (A, 0, len(A)-1)

Daftar = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]

print (bubbleSort(Daftar))
print (selectionSort(Daftar))
print (insertionSort(Daftar))
mergeSort(Daftar)
print (Daftar)
quickSort(Daftar)
print (Daftar)

k =  [[i] for i in range(1, 6001)]
kocok(k)
u_bub = k[:]
u_sel = k[:]
u_ins = k[:]
u_mrg = k[:]
u_qck = k[:]

aw=detak();bubbleSort(u_bub);ak=detak();print("bubble: %g detik" %(ak-aw));
aw=detak();selectionSort(u_sel);ak=detak();print("selection: %g detik" %(ak-aw))
aw=detak();insertionSort(u_ins);ak=detak();print("insertion: %g detik" %(ak-aw))
aw=detak();mergeSort(u_mrg);ak=detak();print("merge: %g detik" %(ak-aw));
aw=detak();quickSort(u_qck);ak=detak();print("quick: %g detik" %(ak-aw));
```

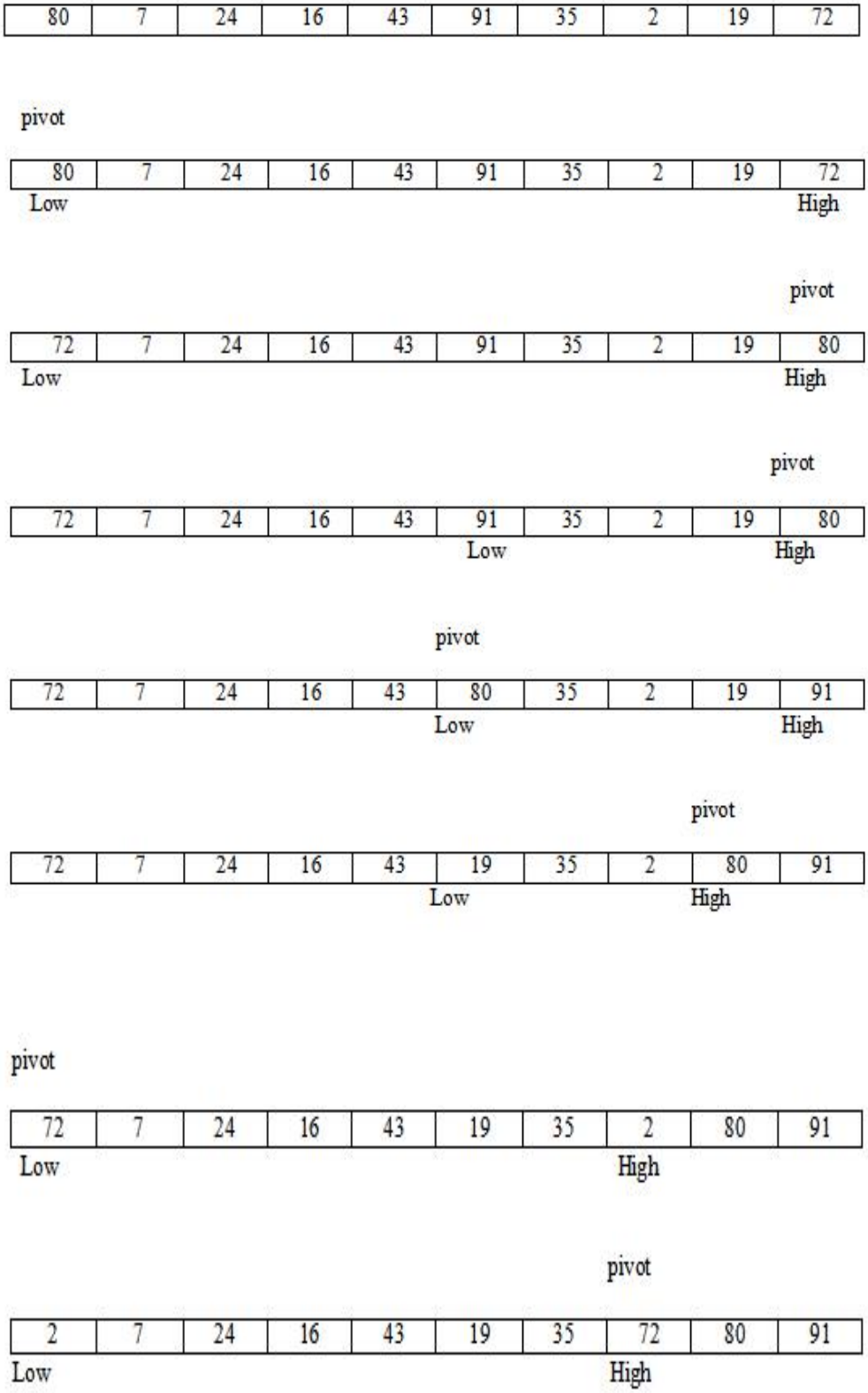
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Ayudhia/Documents/SEMESTER 4/3.py ==========
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
[2, 4, 5, 10, 13, 18, 23, 29, 31, 51, 64]
bubble: 12.3961 detik
selection: 5.20503 detik
insertion: 5.35933 detik
merge: 0.0831068 detik
quick: 0.0642447 detik
>>>
```

4. Diberikan list L = [80, 7, 24, 16, 43, 91, 35, 2, 19, 72], gambarlah trace pengurutan untuk algoritma.

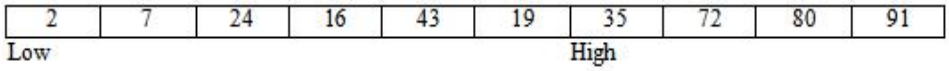
a) Merge Sort



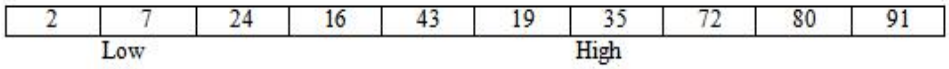
b) Quick Sort


80 7 24 16 43 91 35 2 19 72
pivot
80 7 24 16 43 91 35 2 19 72
Low
High
pivot
72 7 24 16 43 91 35 2 19 80
Low
High
pivot
72 7 24 16 43 91 35 2 19 80
Low
High
pivot
72 7 24 16 43 80 35 2 19 91
Low
High
pivot
72 7 24 16 43 19 35 2 80 91
Low
High
pivot
72 7 24 16 43 19 35 2 80 91
Low
High
pivot
2 7 24 16 43 19 35 72 80 91
Low
High

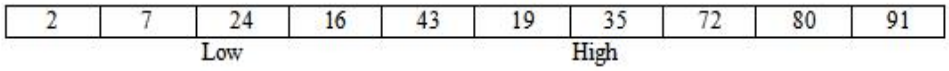
pivot



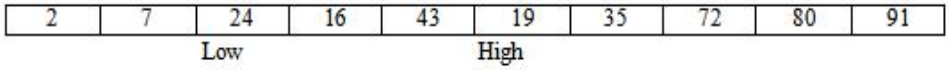
pivot



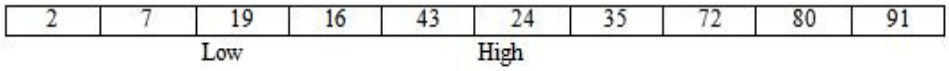
pivot



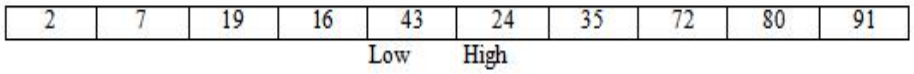
pivot



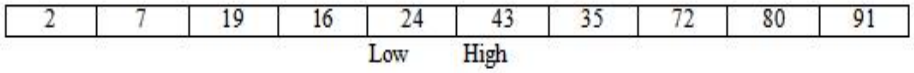
pivot



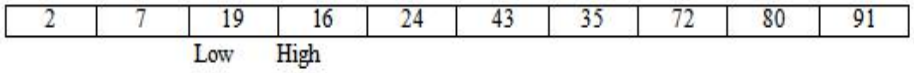
pivot



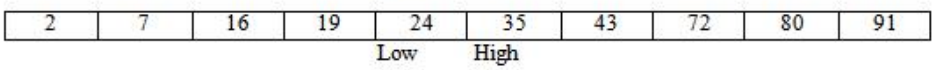
pivot

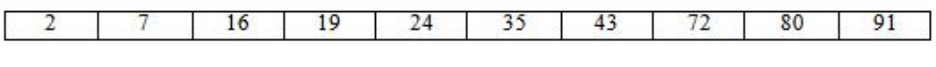


pivot

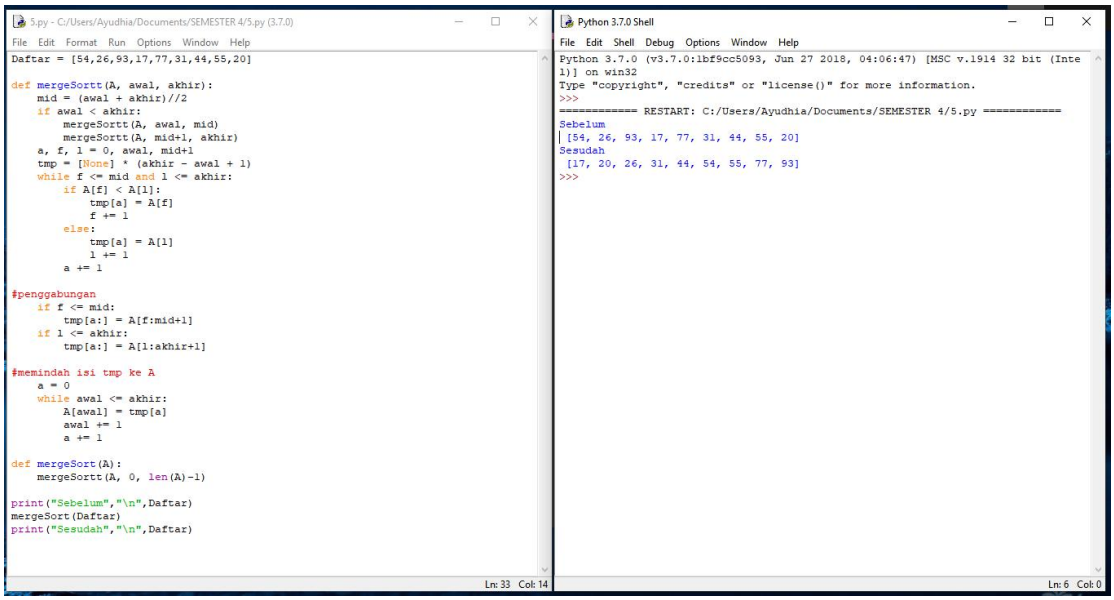


pivot





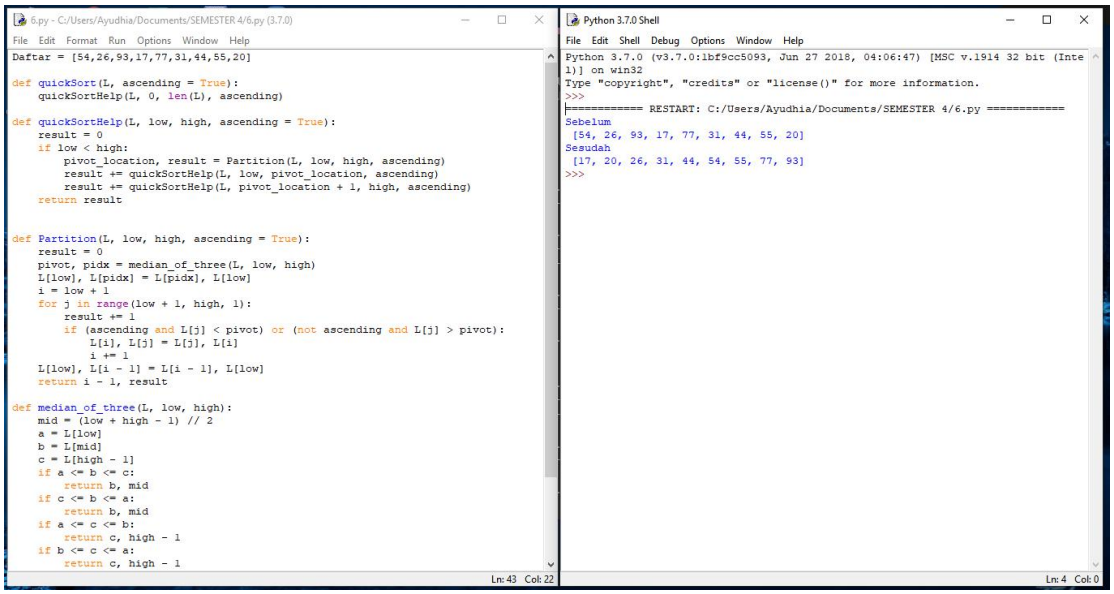
5. Tingkatkan efiesiensi program mergeSort dengan tidak memakai operator slice.



```
Daftar = [54,26,93,17,77,31,44,55,20]

def mergeSortt(A, awal, akhir):
    mid = (awal + akhir)//2
    if awal < akhir:
        mergeSortt(A, awal, mid)
        mergeSortt(A, mid+1, akhir)
    a, f, l = 0, awal, mid+1
    tmp = [None] * (akhir - awal + 1)
    while f <= mid and l <= akhir:
        if A[f] < A[l]:
            tmp[a] = A[f]
            f += 1
        else:
            tmp[a] = A[l]
            l += 1
        a += 1

#penggabungan
    if f <= mid:
        tmp[a:] = A[f:mid+1]
    if l <= akhir:
        tmp[a:] = A[l:akhir+1]

#memindah isi tmp ke A
    a = 0
    while awal <= akhir:
        A[awal] = tmp[a]
        awal += 1
        a += 1

def mergeSort(A):
    mergeSortt(A, 0, len(A)-1)

print("Sebelum","\n",Daftar)
mergeSort(Daftar)
print("Sesudah","\n",Daftar)
```

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Ayudhia/Documents/SEMESTER 4/5.py ==========
Sebelum
 [54, 26, 93, 17, 77, 31, 44, 55, 20]
Sesudah
 [17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
```

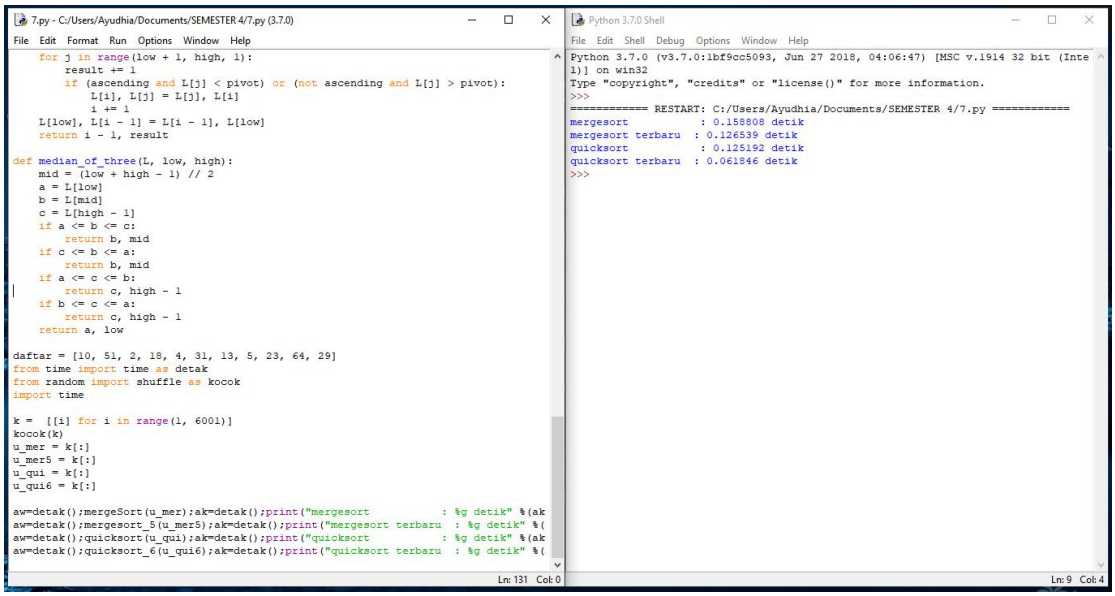
6. Ubahlah kodenya dan ujilah.



```
Daftar = [54,26,93,17,77,31,44,55,20]

def quickSort(L, ascending = True):
    quickSortHelp(L, 0, len(L), ascending)

def quickSortHelp(L, low, high, ascending = True):
    result = 0
    if low < high:
        pivot_location, result = Partition(L, low, high, ascending)
        result += quickSortHelp(L, low, pivot_location, ascending)
        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
```

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Ayudhia/Documents/SEMESTER 4/6.py ==========
Sebelum
 [54, 26, 93, 17, 77, 31, 44, 55, 20]
Sesudah
 [17, 20, 26, 31, 44, 54, 55, 77, 93]
>>>
```

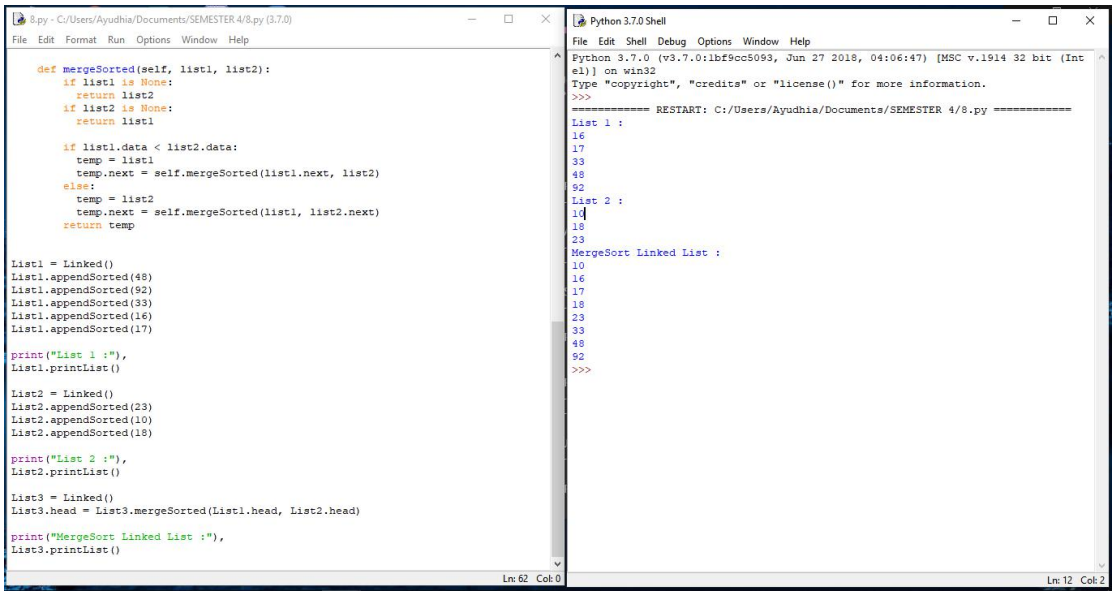
7. Uji kecepatan keduanya dan perbandingkan juga dengan kode awalnya.



```
7.py - C:/Users/Ayudhia/Documents/SEMESTER 4/7.py (3.7.0)
File  Edit  Format  Run  Options  Window  Help
    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

daftar = [10, 51, 2, 18, 4, 31, 13, 5, 23, 64, 29]
from time import time as detak
from random import shuffle as kocok
import time

k =  [[i] for i in range(1, 6001)]
kocok(k)
u_mer = k[:]
u_mer5 = k[:]
u_qui = k[:]
u_qui6 = k[:]

aw=detak();mergeSort(u_mer);ak=detak();print("mergesort           : %g detik" %(ak
aw=detak();mergesort_5(u_mer5);ak=detak();print("mergesort terbaru  : %g detik" %(
aw=detak();quicksort(u_qui);ak=detak();print("quicksort           : %g detik" %(ak
aw=detak();quicksort_6(u_qui6);ak=detak();print("quicksort terbaru  : %g detik" %(
Ln: 131  Col: 0
```

```
Python 3.7.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
============ RESTART: C:/Users/Ayudhia/Documents/SEMESTER 4/7.py ============
mergesort           : 0.158808 detik
mergesort terbaru  : 0.126539 detik
quicksort           : 0.125192 detik
quicksort terbaru  : 0.061846 detik
>>>
Ln: 9  Col: 4
```

8. Buatlah versi linked list untuk program mergeSort di atas.



```
8.py - C:/Users/Ayudhia/Documents/SEMESTER 4/8.py (3.7.0)
File  Edit  Format  Run  Options  Window  Help
    def mergeSorted(self, list1, list2):
        if list1 is None:
          return list2
        if list2 is None:
          return list1

        if list1.data < list2.data:
          temp = list1
          temp.next = self.mergeSorted(list1.next, list2)
        else:
          temp = list2
          temp.next = self.mergeSorted(list1, list2.next)
        return temp


List1 = Linked()
List1.appendSorted(48)
List1.appendSorted(92)
List1.appendSorted(33)
List1.appendSorted(16)
List1.appendSorted(17)

print("List 1 :"),
List1.printList()

List2 = Linked()
List2.appendSorted(23)
List2.appendSorted(10)
List2.appendSorted(18)

print("List 2 :"),
List2.printList()

List3 = Linked()
List3.head = List3.mergeSorted(List1.head, List2.head)

print("MergeSort Linked List :"),
List3.printList()
Ln: 62  Col: 0
```

```
Python 3.7.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Int
el)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
============ RESTART: C:/Users/Ayudhia/Documents/SEMESTER 4/8.py ============
List 1 :
16
17
33
48
92
List 2 :
10
18
23
MergeSort Linked List :
10
16
17
18
23
33
48
92
>>>
Ln: 12  Col: 2
```