Nama: Sina Pijar Sahmura

Nim: 2400606

Kelas: 1B

* Dari algoritma yang telah anda pilih pada studi kasus 1, buatlah perbandinganrunning program (execution time) dengan 2 algoritma yang ada di slide (pilih: bubble sort, insertion sort, atau selection sort, quick sort, merge sort, radix sort counting sort).
* Buatlah dengan menggunakan array acak berikut:

[7, 1, 36, 26, 63, 93, 55, 16, 19, 38, 74, 65, 18, 59, 8, 43, 24, 79, 49, 35, 23, 78, 51, 2, 46, 28, 60, 76, 10, 85, 66, 29, 82, 58, 69, 75, 48, 100, 5, 32, 40, 33, 34, 90, 81, 42, 57, 44, 41, 77]

* Tentukan manakah dari ke 3 algoritma tersebut yang paling cepat?
* Kumpulkan file dalam bentuk copy kodingan dan screenshot hasil program dan simpan ke dalam file dengan format .pdf

Jawaban:

'''

Nama: Sina Pijar Sahmura

Nim: 2400606

Kelas: 1B

'''

print("Merge Sort")

def merge\_sort(arr):

    if len(arr) <= 1:

        return arr

    mid = len(arr) // 2

    leftHalf = arr[:mid]

    rightHalf = arr[mid:]

    sortedleft = merge\_sort(leftHalf)

    sortedright = merge\_sort(rightHalf)

    return merge(sortedleft, sortedright)

def merge(left, right):

    result = []

    i = j = 0

    while i < len(left) and j < len(right):

        if left[i] < right[j]:

            result.append(left[i])

            i += 1

        else:

            result.append(right[j])

            j += 1

    result.extend(left[i:])

    result.extend(right[j:])

    return result

unsortedArr = [7, 1, 36, 26, 63, 93, 55, 16, 19, 38, 74, 65, 18,

               59, 8, 43, 24, 79, 49, 35, 23, 78, 51, 2, 46, 28,

               60, 76, 10, 85, 66, 29, 82, 58, 69, 75, 48, 100,

               5, 32, 40, 33, 34, 90, 81, 42, 57, 44, 41, 77]

sortedArr = merge\_sort(unsortedArr)

print(f"sebelum diurutkan {unsortedArr}")

print(f"Setela diurutkan {sortedArr}")

'''

Nama: Sina Pijar Sahmura

Nim: 2400606

Kelas: 1B

'''

print("Counting Sort")

def counting\_sort(arr):

    max\_val = max(arr)

    min\_val = min(arr)

    count = [0] \* (max\_val - min\_val + 1)

    for num in arr:

        count[num - min\_val] += 1

    arr[:] = [i + min\_val for i, c in enumerate(count) for \_ in range(c)]

arr = [7, 1, 36, 26, 63, 93, 55, 16, 19, 38, 74, 65, 18,

       59, 8, 43, 24, 79, 49, 35, 23, 78, 51, 2, 46, 28,

       60, 76, 10, 85, 66, 29, 82, 58, 69, 75, 48, 100,

       5, 32, 40, 33, 34, 90, 81, 42, 57, 44, 41, 77]

print(f"Sebelum diurutkan: {arr}")

counting\_sort(arr)

print(f"Setelah diurutkan: {arr}")

'''

Nama: Sina Pijar Sahmura

Nim: 2400606

Kelas: 1B

'''

print("Quick Sort")

def quick\_sort(arr):

    if len(arr) <= 1:

        return arr

    pivot = arr[len(arr) // 2]

    left = [x for x in arr if x < pivot]

    middle = [x for x in arr if x == pivot]

    right = [x for x in arr if x > pivot]

    return quick\_sort(left) + middle + quick\_sort(right)

unsortedarr = [7, 1, 36, 26, 63, 93, 55, 16, 19, 38, 74, 65, 18,

               59, 8, 43, 24, 79, 49, 35, 23, 78, 51, 2, 46, 28,

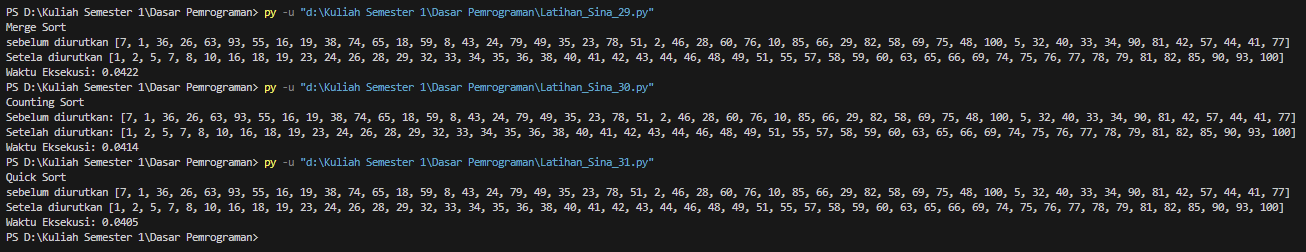
               60, 76, 10, 85, 66, 29, 82, 58, 69, 75, 48, 100,

               5, 32, 40, 33, 34, 90, 81, 42, 57, 44, 41, 77]

print(f"sebelum diurutkan {unsortedarr}")

sortedarr = quick\_sort(unsortedarr)

print(f"Setela diurutkan {sortedarr}")



Berdasarkan hasil pengujian waktu eksekusi, Quick Sort merupakan algoritma yang paling cepat di antara ketiga algoritma yang diuji.