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}")
```

```
PS D:\Kuliah Semester 1\Dasar Pemrograman> py -u "d:\Kuliah Semester 1\Dasar Pemrograman\Latihan_Sina_29.py"
Merge Sort
sebelum diurutkan [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]
Setela diurutkan [1, 2, 5, 7, 8, 10, 16, 18, 19, 23, 24, 26, 28, 29, 32, 33, 34, 35, 36, 38, 40, 41, 42, 43, 44, 46, 48, 49, 51, 55, 57, 58, 59, 60, 63, 65, 66, 69, 74, 75, 76, 77, 78, 79, 81, 82, 85, 90, 93, 100]
Waktu Eksekusi: 0.0422
PS D:\Kuliah Semester 1\Dasar Pemrograman> py -u "d:\Kuliah Semester 1\Dasar Pemrograman\Latihan_Sina_30.py"
Counting Sort
Sebelum diurutkan: [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]
Setelah diurutkan: [1, 2, 5, 7, 8, 10, 16, 18, 19, 23, 24, 26, 28, 29, 32, 33, 34, 35, 36, 38, 40, 41, 42, 43, 44, 46, 48, 49, 51, 55, 57, 58, 59, 60, 63, 65, 66, 69, 74, 75, 76, 77, 78, 79, 81, 82, 85, 90, 93, 100]
Waktu Eksekusi: 0.0414
PS D:\Kuliah Semester 1\Dasar Pemrograman> py -u "d:\Kuliah Semester 1\Dasar Pemrograman\Latihan_Sina_31.py"
Quick Sort
sebelum diurutkan [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]
Setela diurutkan [1, 2, 5, 7, 8, 10, 16, 18, 19, 23, 24, 26, 28, 29, 32, 33, 34, 35, 36, 38, 40, 41, 42, 43, 44, 46, 48, 49, 51, 55, 57, 58, 59, 60, 63, 65, 66, 69, 74, 75, 76, 77, 78, 79, 81, 82, 85, 90, 93, 100]
Waktu Eksekusi: 0.0405
PS D:\Kuliah Semester 1\Dasar Pemrograman>
```

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