2 sets of python code that work for quick sort:

Complex and lots of code way

import csv

import sys

import operator

import pandas as pd

import time

# Driver code to test above

starttime = time.time()

path = 'C:/Users/razzb/Desktop/quicksort/Dateset.csv'

file\_CSV = open(path,'r')

data = pd.read\_csv(file\_CSV)

test = data.sort\_values("Age", ascending=True, inplace=False, kind='quicksort', na\_position='last')

print(test)

print("elapsed time is:")

print(time.time()-starttime)

# Python program for implementation of Quicksort Sort

# This function takes last element as pivot, places

# the pivot element at its correct position in sorted

# array, and places all smaller (smaller than pivot)

# to left of pivot and all greater elements to right

# of pivot

import csv

import sys

import operator

import time

sys.setrecursionlimit(5000000)

def partition(arr, low, high):

    i = (low-1)         # index of smaller element

    pivot = arr[high]     # pivot

    for j in range(low, high):

        # If current element is smaller than or

        # equal to pivot

        if arr[j] <= pivot:

            # increment index of smaller element

            i = i+1

            arr[i], arr[j] = arr[j], arr[i]

    arr[i+1], arr[high] = arr[high], arr[i+1]

    return (i+1)

# The main function that implements QuickSort

# arr[] --> Array to be sorted,

# low  --> Starting index,

# high  --> Ending index

# Function to do Quick sort

def quickSort(arr, low, high):

    if len(arr) == 1:

        return arr

    if low < high:

        # pi is partitioning index, arr[p] is now

        # at right place

        pi = partition(arr, low, high)

        # Separately sort elements before

        # partition and after partition

        quickSort(arr, low, pi-1)

        quickSort(arr, pi+1, high)

# Driver code to test above

start\_time = time.time()

path = 'C:/Users/razzb/Desktop/quicksort/Dateset.csv'

file\_CSV = open(path,'r')

data\_CSV = csv.reader(file\_CSV)

arr= list(data\_CSV)

arr2= list(data\_CSV)

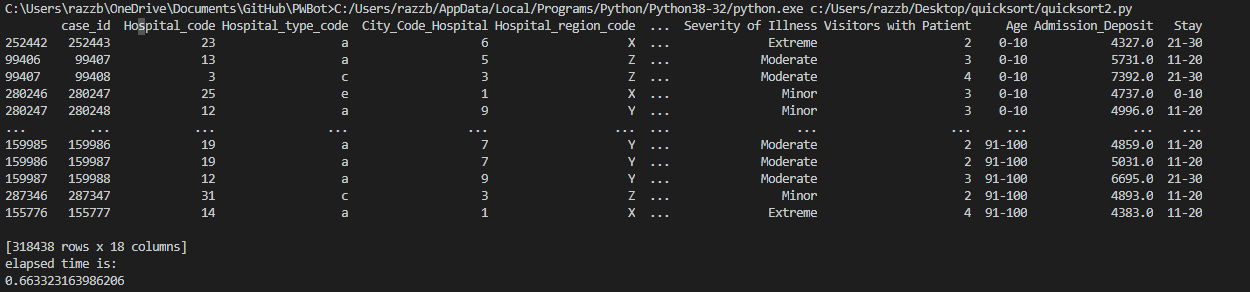
n = len(arr)

quickSort(arr, 0, n-1)

print( time.time() - start\_time )

# This code is contributed by Mohit Kumra

#This code in improved by https://github.com/anushkrishnav

Elapsed time for quicksort is roughly half a second. 0.6565971374511719