# Bubble sort

import random

import time

from datetime import datetime

def bubble\_sort(arr):

n = len(arr)

for i in range(n):

for j in range(n-1):

if arr[j] > arr[j+1]:

arr[j], arr[j+1] = arr[j+1], arr[j]

def main():

a = [random.randint(0, 10000) for \_ in range(10000)]

start = datetime.now()

bubble\_sort(a)

end = datetime.now()

duration = end - start

print("Serial:", duration)

a = [random.randint(0, 10000) for \_ in range(10000)]

start = datetime.now()

for i in range(len(a)):

first = i % 2

for j in range(first, len(a)-1, 2):

if a[j] > a[j+1]:

a[j], a[j+1] = a[j+1], a[j]

end = datetime.now()

duration = end - start

print("Parallel:", duration)

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Merge sort

import concurrent.futures

import time

def merge(a, i1, j1, i2, j2):

temp = []

i, j = i1, i2

while i <= j1 and j <= j2:

temp.append(a[i] if a[i] < a[j] else a[j])

i, j = (i + 1, j) if a[i] < a[j] else (i, j + 1)

temp.extend(a[i: j1 + 1])

temp.extend(a[j: j2 + 1])

a[i1: j2 + 1] = temp

def mergesort(a, i, j):

if i < j:

mid = (i + j) // 2

with concurrent.futures.ThreadPoolExecutor() as executor:

future1 = executor.submit(mergesort, a, i, mid)

future2 = executor.submit(mergesort, a, mid + 1, j)

future1.result()

future2.result()

merge(a, i, mid, mid + 1, j)

if \_\_name\_\_ == "\_\_main\_\_":

a = [3,44,38,5,47,15,36,26,27,2,46,4,19,1,50,48]

start\_p=time.time()

mergesort(a, 0, len(a) - 1)

end\_p=time.time()

print("Sorted array is:")

for element in a:

print(element)

print("Execution time:", end\_p-start\_p, "seconds")