

情報処理 III

後期 第3回課題

4D38 宮里 孝希

2024/10/04

プログラム

SORTING ALGORITHMS

```
def normal_quick(a):
    length = len(a)
    stack = [(0, length - 1)]
    while len(stack) > 0:
        left, right = stack.pop()
        if left >= right:
            continue
        i = left
        j = right
        s = a[left + (right - left) // 2]
        while i < j:
            while a[i] < s:
                i += 1
            while a[j] > s:
                j -= 1
            if i < j:
                if a[i] == a[j]:
                    j -= 1
                else:
                    a[i], a[j] = a[j], a[i]
        stack.append((j + 1, right))
        stack.append((left, i - 1))
```

```
def recursive_quick(arr, left, right):
    if left < right:
        pivot = arr[left]
        low = left
        high = right + 1

        while True:
            low += 1
            while low <= right and arr[low] < pivot:
                low += 1
            high -= 1
            while arr[high] > pivot:
                high -= 1
            if low >= high:
                break

            # Swap elements at low and high
            arr[low], arr[high] = arr[high], arr[low]

        # Place pivot in the correct position
        arr[left], arr[high] = arr[high], arr[left]

        # Recursively sort the left and right subarrays
        recursive_quick(arr, left, high - 1)
        recursive_quick(arr, high + 1, right)
```

```

##### MAIN #####

import random
import time

reursive_time = []
normal_time = []

for trial in range(10):
    print(trial + 1, "回目")

    rand = [] # 空のリストを作成

    for i in range(1000000): # 乱数の個数を指定
        rand.append(random.randint(1, 1000000))

    recursive_start = time.time()
    recursive_quick(rand, 0, len(rand) - 1)
    recursive_end = time.time()
    reursive_time.append(recursive_end - recursive_start)
    print("再帰関数:", rand[:4], "時間:", reursive_time[trial])

    normal_start = time.time()
    normal_quick(rand)
    normal_end = time.time()
    normal_time.append(normal_end - normal_start)
    print("通常関数:", rand[:4], "時間:", normal_time[trial])
    print("")

print("### result ###")
print("再帰関数 (平均) :", sum(reursive_time) / len(reursive_time))
print("通常関数 (平均) :", sum(normal_time) / len(normal_time))

```

実行結果

1 回目

再帰関数: [4, 4, 14, 17] 時間: 0.9432158470153809

通常関数: [4, 4, 14, 17] 時間: 0.5442979335784912

2 回目

再帰関数: [10, 13, 33, 51] 時間: 0.942119836807251

通常関数: [10, 13, 33, 51] 時間: 0.5192341804504395

3 回目

再帰関数: [9, 37, 45, 69] 時間: 0.9363369941711426

通常関数: [9, 37, 45, 69] 時間: 0.5365657806396484

4 回目

再帰関数: [15, 20, 25, 50] 時間: 0.9245579242706299

通常関数: [15, 20, 25, 50] 時間: 0.5295391082763672

5 回目

再帰関数: [6, 7, 13, 18] 時間: 0.9399800300598145

通常関数: [6, 7, 13, 18] 時間: 0.5212030410766602

6 回目

再帰関数: [7, 7, 13, 15] 時間: 0.9128258228302002

通常関数: [7, 7, 13, 15] 時間: 0.5382277965545654

7 回目

再帰関数: [1, 11, 14, 15] 時間: 0.9354259967803955

通常関数: [1, 11, 14, 15] 時間: 0.5240631103515625

8 回目

再帰関数: [8, 11, 14, 15] 時間: 0.9203028678894043

通常関数: [8, 11, 14, 15] 時間: 0.5371153354644775

9 回目

再帰関数: [11, 16, 19, 29] 時間: 0.948451042175293

通常関数: [11, 16, 19, 29] 時間: 0.5292460918426514

10 回目

再帰関数: [20, 32, 37, 38] 時間: 0.9280388355255127

通常関数: [20, 32, 37, 38] 時間: 0.5417788028717041

result

再帰関数 (平均) : 0.9331255197525025

通常関数 (平均) : 0.5321271181106567
