

Quick Sort Kuy

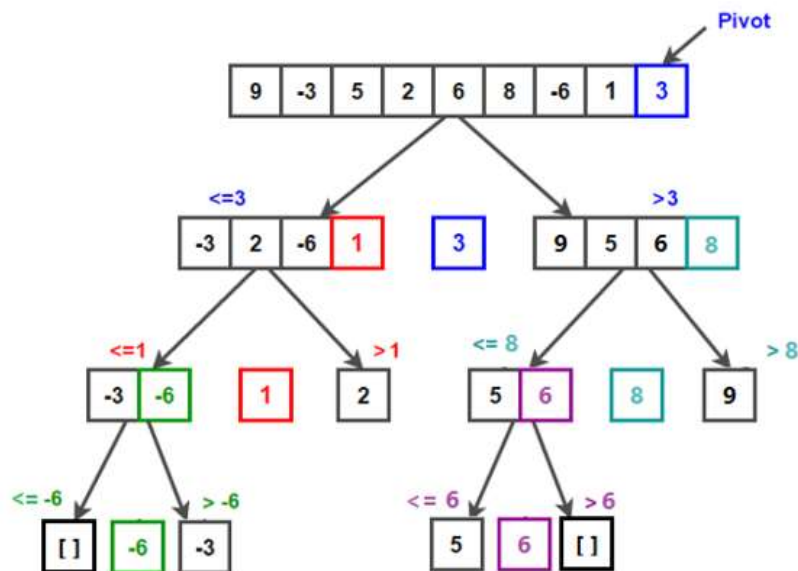
Problem

Submissions

Leaderboard

Discussions

Quick sort merupakan algoritma sorting yang paling umum digunakan saat ini. Algoritma ini merupakan salah satu implementasi konsep Divide and Conquer, dimana objek dibagi menjadi bagian kecil secara berulang-ulang hingga mencapai bagian paling sederhana:



Pada kasus quicksort terhadap array of integer, akan diambil sebuah pivot yang merupakan salah satu elemen dari array (pivot dapat diambil dari elemen manapun). Array akan dipecah menjadi 2 bagian, yaitu array dengan elemen kurang dari pivot (sebut saja array kiri) dan lebih dari pivot (array kanan). Proses ini diulang lagi pada kedua array yang baru (pada array kiri dan array kanan) hingga mencapai bagian paling kecil (basis), yaitu dimana ukuran array sama dengan 1. Setelah itu, array kiri, pivot, dan array kanan disatukan kembali menjadi array yang telah terurut (conquer).

Untuk membantu menambah pemahaman mengenai Divide and Conquer, implementasikan algoritma quicksort, dengan mengambil **pivot elemen paling akhir (seperti gambar)**, array kiri berisikan elemen \leq pivot, dan array kanan berisikan elemen $>$ pivot. Hasil array kiri dan array kanan sesuai dengan urutan parent.

Contoh branching:

- Apabila parent: 5, 2, 1, 4, 3
- Pivot: 3 (elemen terakhir dari parent yang ada)

- Left array (≤ 3): 2, 1 (5, **2**, 1, 4, 3)

- Right array (> 3): 5, 4 (**5**, 2, 1, **4**, 3)

Pada setiap tahap pemecahan array, outputkan pivot, array kiri (*sebelum disort / hasil dipisah pivot saja*), dan array kanan (*sebelum disort / hasil dipisah pivot saja*) yang dihasilkan secara infix. Setelah itu, outputkan array yang telah tersortir

Input Format

Baris pertama berisikan satu buah integer yang merupakan panjang array (n)

Baris kedua berisikan n buah integer yang merupakan elemen-elemen dari array (arr[i]).

Constraints

$1 \leq n \leq 1000$

$-1000 \leq \text{arr}[i] \leq 1000$

Output Format

Outputkan:

Pivot: [pivot yang digunakan]

Left arr: [array kiri, dipisahkan spasi]

Right arr: [array kanan, dipisahkan spasi]

untuk setiap branching. Urutan **prefix** (print, branch ke kiri, branch ke kanan, lihat sample case). Bila ukuran array hanya satu, **tidak perlu diprint pivot, left array, dan right arraynya**. Anggap array dengan ukuran ≤ 1 sebagai basis

Pada baris terakhir, outputkan:

Final: [array akhir yang terurut]

Sample Input 0

```
9
9 -3 5 2 6 8 -6 1 3
```

Sample Output 0

```
Pivot: 3
Left arr: -3 2 -6 1
Right arr: 9 5 6 8
```

```
Pivot: 1
Left arr: -3 -6
Right arr: 2
```

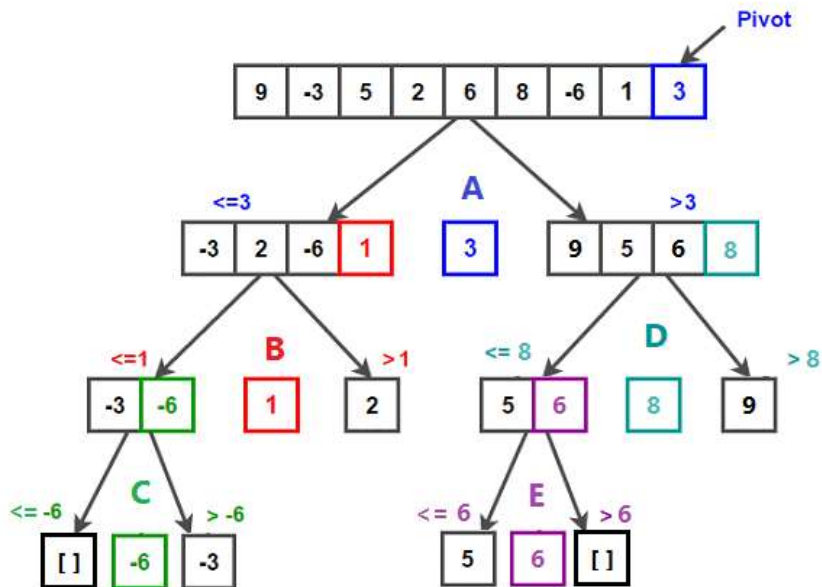
```
Pivot: -6
Left arr:
Right arr: -3
```

```
Pivot: 8
Left arr: 5 6
Right arr: 9
```

```
Pivot: 6
Left arr: 5
Right arr:
```

```
Final: -6 -3 1 2 3 5 6 8 9
```

Explanation 0



Prefix:

1. Print
2. Branch left
3. Branch right

Note: GIF dapat di screenshot apabila terasa terlalu cepat

f t in

Contest ends in 2 days

Submissions: 70

Max Score: 100

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

[More](#)

```

1 def quicksort(arr):
2     n = len(arr)
3     if n <= 1:
4         return arr
5     else:
6         pivot = arr[n - 1]
7         left_arr = [x for x in arr[:n - 1] if x <= pivot]
8         right_arr = [x for x in arr[:n - 1] if x > pivot]
9
10        print("Pivot:", pivot)
11        print("Left arr:", " ".join(str(x) for x in left_arr))
12        print("Right arr:", " ".join(str(x) for x in right_arr))
13        print()
14

```

```
15         left_sorted = quicksort(left_arr)
16         right_sorted = quicksort(right_arr)
17
18         return left_sorted + [pivot] + right_sorted
19
20 n = int(input())
21 arr = list(map(int, input().split()))
22
23 result = quicksort(arr)
24
25 print("Final:", " ".join(str(x) for x in result))
```

Line: 1 Col: 1



[Upload Code as File](#)

☐

Test against custom input

Run Code

Submit Code