

1.

```
StackQueue_G211220038.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at December 10

+ Code + Text

# Queue
from collections import deque

queue = deque(["Ram", "Tarun", "Asif", "John"])
print(queue)

queue.append("Akbar")
print(queue)

queue.append("Birbal")
print(queue)

print(queue.popleft())
print(queue.popleft())
print(queue)

# Tower of Hanoi
def TowerOfHanoi(n, source, target, auxiliary):
    if n == 0:
        return
    TowerOfHanoi(n-1, source, auxiliary, target)
    print(f"Move disk {n} from {source} to {target}")
    TowerOfHanoi(n-1, auxiliary, source, target)

# Driver code
N = 3
TowerOfHanoi(N, 'A', 'C', 'B')
```

```
StackQueue_G211220038.ipynb ☆
File Edit View Insert Runtime Tools Help

+ Code + Text

# Sorting in a stack
def sortStack(input_stack):
    tmp_stack = []

    while len(input_stack) > 0:
        tmp = input_stack[-1]
        input_stack.pop()

        while len(tmp_stack) > 0 and tmp_stack[-1] < tmp:
            input_stack.append(tmp_stack[-1])
            tmp_stack.pop()

        tmp_stack.append(tmp)

    return tmp_stack

def sortArrayUsingStacks(arr, n):
    input_stack = []
    i = 0

    while i < n:
        input_stack.append(arr[i])
        i = i + 1

    tmp_stack = sortStack(input_stack)
    i = 0

    while i < n:
        arr[i] = tmp_stack[-1]
        tmp_stack.pop()
        i = i + 1

    return arr
```

```
# Driver code
arr_to_sort = [10, 5, 15, 45]
n_to_sort = len(arr_to_sort)
arr_to_sort = sortArrayUsingStacks(arr_to_sort, n_to_sort)
i = 0

while i < n_to_sort:
    print(arr_to_sort[i], end=" ")
    i = i + 1

deque(['Ram', 'Tarun', 'Asif', 'John'])
deque(['Ram', 'Tarun', 'Asif', 'John', 'Akbar'])
deque(['Ram', 'Tarun', 'Asif', 'John', 'Akbar', 'Birbal'])
Ram
Tarun
deque(['Asif', 'John', 'Akbar', 'Birbal'])
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to A
Move disk 3 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to B
5 10 15 45
```

2. Algoritma dari source code

- Bagian 1: Queue

```
# Queue
from collections import deque

queue = deque(["Ram", "Tarun", "Asif", "John"])
print(queue)

queue.append("Akbar")
print(queue)

queue.append("Birbal")
print(queue)

print(queue.popleft())
print(queue.popleft())
print(queue)
```

Algoritma:

1. Buat deque (antrian) dengan elemen awal "Ram", "Tarun", "Asif", dan "John".
2. Tambahkan "Akbar" dan "Birbal" ke dalam antrian.
3. Hapus dan cetak elemen pertama dari antrian (Ram).
4. Hapus dan cetak elemen kedua dari antrian (Tarun).
5. Cetak antrian setelah dua operasi pop.

- Bagian 2: Tower of Hanoi

```
# Tower of Hanoi
def TowerOfHanoi(n, source, target, auxiliary):
    if n == 0:
        return
    TowerOfHanoi(n-1, source, auxiliary, target)
    print(f"Move disk {n} from {source} to {target}")
    TowerOfHanoi(n-1, auxiliary, source, target)

# Driver code
N = 3
TowerOfHanoi(N, 'A', 'C', 'B')
```

Algoritma:

1. Definisikan fungsi TowerOfHanoi untuk menyelesaikan masalah Tower of Hanoi.
2. Jika n (jumlah disk) sama dengan 0, kembalikan.
3. Pindahkan n-1 disk dari tiang sumber ke tiang bantu.
4. Pindahkan disk ke-n dari tiang sumber ke tiang target.
5. Pindahkan n-1 disk dari tiang bantu ke tiang target.

Gunakan fungsi untuk menyelesaikan masalah Tower of Hanoi dengan N disk dan tiang awal 'A', tiang target 'C', dan tiang bantu 'B'.

- Bagian 3: Sorting in a Stack

```
# Sorting in a stack
def sortStack(input_stack):
    tmp_stack = []

    while len(input_stack) > 0:
        tmp = input_stack[-1]
        input_stack.pop()

        while len(tmp_stack) > 0 and tmp_stack[-1] < tmp:
            input_stack.append(tmp_stack[-1])
            tmp_stack.pop()

        tmp_stack.append(tmp)

    return tmp_stack

def sortArrayUsingStacks(arr, n):
    input_stack = []
    i = 0

    while i < n:
        input_stack.append(arr[i])
        i = i + 1

    tmp_stack = sortStack(input_stack)
    i = 0

    while i < n:
        arr[i] = tmp_stack[-1]
        tmp_stack.pop()
        i = i + 1

    return arr
```

```
# Driver code
arr_to_sort = [10, 5, 15, 45]
n_to_sort = len(arr_to_sort)
arr_to_sort = sortArrayUsingStacks(arr_to_sort, n_to_sort)
i = 0

while i < n_to_sort:
    print(arr_to_sort[i], end=" ")
    i = i + 1
```

Algoritma:

1. Buat fungsi sortStack untuk mengurutkan stack dengan menggunakan stack sementara.
2. Pop elemen dari stack asli dan bandingkan dengan elemen pada stack sementara.
3. Geser elemen-elemen yang lebih kecil ke stack sementara.
4. Tambahkan elemen yang sedang diproses ke stack sementara.
5. Ulangi langkah 2-4 hingga stack asli kosong.
6. Buat fungsi sortArrayUsingStacks untuk mengurutkan array menggunakan stack.
7. Tambahkan elemen array ke stack.
8. Panggil fungsi sortStack untuk mengurutkan stack.
9. Pindahkan elemen-elemen dari stack sementara ke array yang sudah diurutkan.
10. Cetak array yang sudah diurutkan.

