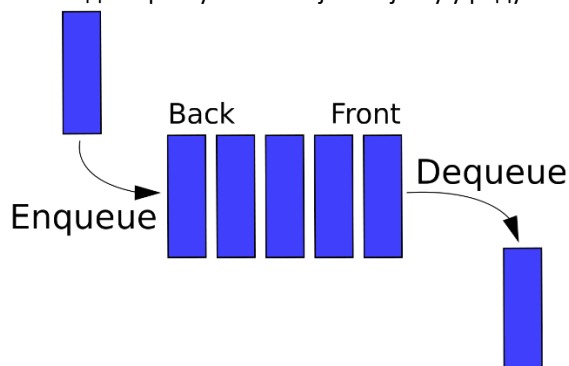


Структура реда у програмирању

Ред је колекција објеката која подржава семантику (FIFO – first-in, first-out) за уметање (enqueue) и одстрањивање (dequeue) података а не подржава слободан приступ било којем објекту у реду осим прво уметнутом објекту у реду.



Редови се користе у разноврсним алгоритмима за решавање проблема распореда и паралелног програмирања.

Алгоритми редоследа често користе интерно редове приоритета.

То су специјализовани редови: уместо добијања следећег елемента по редоследу, они узимају елемент који је највећег приоритета.

Приоритет појединачних елемената се одређује према редоследу премињеном на кључеве елемената.

Обичан ред не мења редослед својих елемената.

У Пајтону су уграђени неколико имплементација реда са мало измењеним карактеристикама.

Ред као имплементација листе

У Пајтону се може лако користити уграђена структура листе као имплементација реда, али због своје мање ефикасности се не користи у алгоритмима.

Ова имплементација се може користити у алгоритмима који захтевају рад са мањим бројем елемената у редовима.

```
q = []
```

```
q.insert(0, 'treći')
```

```
q.insert(0, 'drugi')
```

```
q.insert(0, 'prvi')
```

```
print(q)           #daje ['prvi', 'drugi', 'treći']
```

```
q.pop(2)           #odstranjuje element 'treći'
```

```
print(q)           #daje ['prvi', 'drugi']
```

Ред као имплементација класе collections.deque

Класа deque имплементира ред са два краја који подржава додавање и одстрањивање елемената са оба краја реда.

Пајтонови deque елементи су имплементирани као објекти у двоструко спрегнутој листи чиме имају повећану ефикасност у раду приликом уметања и одстрањивања елемената.

Ипак ова имплементација није ефикасна код приступа елементима у средини реда.

Из свих ових разлога ова имплементација подржава употребу класе deque и за рад са стековима и за рад са редовима.

Често се користи за реализацију структуре података унутар Пајтонове стандардне библиотеке.

```
from collections import deque
```

```
q = deque()
```

```
q.append('prvi')
```

```
q.append('drugi')
```

```
q.append('treći')
```

```
print(q)           #daje deque(['prvi', 'drugi', 'treći'])
```

```
q.popleft()        #odstranjuje 'prvi' element
```

```
q.popleft()        #odstranjuje 'drugi' element
```

```
q.popleft()        #odstranjuje 'treći' element
```

```
q.popleft()        #javlja gresku IndexError: "pop from an empty deque"
```