
Python deque from collections: **Συνοπτική περιγραφή χρήσιμων μεθόδων.**

Επιμέλεια: Α.Δρακόπουλος

Πίνακας περιεχομένων

1. Τι είναι το deque της Python;.....	3
2. Τρόπος χρήσης.....	3
3. Μέθοδοι του deque που χρησιμοποιούνται στην άσκηση BFS.....	3
4. Αναλυτική περιγραφή μεθόδων + παραδείγματα.....	4
4.1 ΜΕΘΟΔΟΣ append(x).....	4
✓ Τι κάνει.....	4
✓ Χρησιμότητα στο BFS.....	4
Παράδειγμα.....	4
Σχόλια.....	4
4.2 ΜΕΘΟΔΟΣ popleft().....	4
✓ Τι κάνει.....	4
✓ Χρησιμότητα στο BFS.....	5
Παράδειγμα.....	5
Σχόλια.....	5
4.3 ΜΕΘΟΔΟΣ appendleft(x) (δεν απαιτείται, αλλά χρήσιμη).....	5
✓ Τι κάνει.....	5
✓ Πού μπορεί να χρησιμοποιηθεί.....	5
Παράδειγμα.....	5
Σχόλια.....	6
4.4 ΜΕΘΟΔΟΣ clear().....	6
✓ Τι κάνει.....	6
✓ Χρήση στο debugging BFS.....	6
Παράδειγμα.....	6
Σχόλια.....	6
Συνοπτικά.....	6

1. Τι είναι το deque της Python;

Το deque (double-ended queue) είναι μια **διπλής κατεύθυνσης ουρά**, υλοποιημένη σε **βέλτιστες δομές μνήμης (doubly linked blocks)** και προσφέρεται από το module `collections`.

Είναι **ιδανική για BFS**, γιατί:

- επιτρέπει **$O(1)$** εισαγωγή στοιχείων στο τέλος $\rightarrow$ `append()`
- επιτρέπει **$O(1)$** αφαίρεση του πρώτου στοιχείου $\rightarrow$ `popleft()`
- δηλαδή υλοποιεί **πραγματική FIFO** ουρά

Στην BFS η FIFO ουρά είναι **απαραίτητη**, γιατί εξασφαλίζει level-order traversal.

2. Τρόπος χρήσης

Για να χρησιμοποιήσουμε deque, γράφουμε:

```
from collections import deque
```

```
q = deque()  
q.append(start)    # εισαγωγή στο τέλος της ουράς  
node = q.popleft() # αφαίρεση από την αρχή
```

Αυτό επιτρέπει:

- σωστή σειρά διερεύνησης των κόμβων
 - αποφυγή κόστους $O(n)$ που έχει η list όταν κάνεις `pop(0)`
-

3. Μέθοδοι του deque που χρησιμοποιούνται στην άσκηση BFS

Στην εργασία χρειάζονται ουσιαστικά **τρεις μέθοδοι**:

1. **`append()`** $\rightarrow$ εισάγει κόμβο στην ουρά
2. **`popleft()`** $\rightarrow$ εξάγει κόμβο από την αρχή της ουράς
3. **`appendleft()`** (προαιρετική, δεν απαιτείται στο BFS)
4. **`clear()`** (χρήσιμη για debugging)

Η βασική BFS χρησιμοποιεί **`append` + `popleft`**.

4. Αναλυτική περιγραφή μεθόδων + παραδείγματα

Παρακάτω αναφέρονται χρήσιμες μέθοδοι με παραδείγματα.

4.1 ΜΕΘΟΔΟΣ `append(x)`

✓ Τι κάνει

Προσθέτει ένα στοιχείο **στο τέλος** της ουράς (δεξιά πλευρά).

✓ Χρησιμότητα στο BFS

Προσθέτει νέους κόμβους που ανακαλύπτουμε στη BFS ώστε να εξερευνηθούν **μετά** τους προηγούμενους.

Παράδειγμα

```
from collections import deque

# Δημιουργία ουράς BFS
q = deque()

# Προσθήκη αρχικού κόμβου
q.append((0, 0)) # το start

print(q)
```

Σχόλια

- Η `append()` χρησιμοποιείται κάθε φορά που βρίσκουμε έναν νέο κόμβο (`neighbor`) στη BFS.
 - Είναι $O(1)$ χρόνος, ανεξάρτητα από το μέγεθος της ουράς.
-

4.2 ΜΕΘΟΔΟΣ `popleft()`

✓ Τι κάνει

Αφαιρεί και επιστρέφει **το πρώτο στοιχείο** της ουράς.

✓ Χρησιμότητα στο BFS

Εξασφαλίζει FIFO → first in, first out

Άρα η BFS εξερευνά κόμβους κατά επίπεδα.

Παράδειγμα

```
from collections import deque
```

```
q = deque()  
q.append("A")  
q.append("B")  
q.append("C")
```

```
x = q.popleft() # returns "A"  
y = q.popleft() # returns "B"
```

```
print(x, y)  
print(q) # περιέχει μόνο "C"
```

Σχόλια

- Η `popleft` είναι το κλειδί της BFS.
 - Αν χρησιμοποιήσεις `pop(0)` από λίστα, το κόστος είναι $O(n)$ → η BFS γίνεται αργή και λανθασμένη στα μεγάλα grids.
-

4.3 ΜΕΘΟΔΟΣ `appendleft(x)` (δεν απαιτείται, αλλά χρήσιμη)

✓ Τι κάνει

Βάζει ένα στοιχείο στην αρχή της ουράς.

✓ Πού μπορεί να χρησιμοποιηθεί

- σε debugging
- σε αλγορίθμους όπου χρειάζεται “front priority”
- σε IDA*, DLS ή 0-1 BFS

Στην εργασία BFS **δεν την χρησιμοποιούμε**, αλλά αναφέρεται για πληρότητα.

Παράδειγμα

```
from collections import deque
```

```
q = deque([2, 3, 4])  
q.appendleft(1)  
print(q) # deque([1, 2, 3, 4])
```

Σχόλια

- Δεν έχει ρόλο στην κλασική BFS.
 - Αλλά βοηθά σε priority-like extensions.
-

4.4 ΜΕΘΟΔΟΣ `clear()`

✓ Τι κάνει

Αδειάζει εντελώς την ουρά.

✓ Χρήση στο **debugging BFS**

Μετά από failed BFS run, μπορείς να καθαρίσεις την ουρά.

Παράδειγμα

```
from collections import deque

q = deque([1, 2, 3])
q.clear()
print(q) # deque([])
```

Σχόλια

- Δεν απαιτείται στην εργασία, αλλά χρήσιμο σε test scripts.
-

Συνοπτικά

1. Το deque είναι **ουρά FIFO** υψηλής απόδοσης.
2. Οι βασικές μέθοδοι για BFS είναι:
 - `append(x)`
 - `popleft()`
3. Χρησιμοποιείται για **level-order traversal** → shortest path.
4. Η BFS χωρίς deque είναι είτε λάθος είτε πολύ αργή.
5. Το parent + visited + deque είναι **ο πυρήνας** της BFS.