

Es stack

lunedì 6 giugno 2022 16:33

Eliminare tutte le occorrenze di k, e le liste di appoggio possono essere solo stack

Cancella(S, k):

```
While not stackEmpty(S):  
    R = pop(s)  
    If r != k:  
        Push(SA, r)  
While not stackEmpty(SA):  
    R = pop(SA)  
    Push(S, r)
```

$$T(n) = 3tw1 + 1tif + 3tw2 = 3n + tif + 3tw2$$

Caso migliore:

- S contiene solo il valore k
Tif = tw2 = 0
 $T(n) = 3n = \Omega(N)$

Caso peggiore:

- K non è mai dentro S
Tif = tw2 = tw1 = n
 $T(n) = 3n + n + 3n = 7n = O(n)$

$$\rightarrow O(n) = \Omega(n) \rightarrow \theta(n)$$

Ricerca un valore K in P

Boolean Trova(P, k):

```
Trovato = false  
While not stackEmpty(P) and not trovato:  
    R = pop(P)  
    If r == k:  
        Trovato = true  
Return trovato
```

Ribalta S

Ribalta(S):

```
While not stackEmpty(S):  
    R = pop(s)  
    If r != k:  
        Push(SA, r)  
While not stackEmpty(SA):  
    R = pop(SA)  
    If r != k:  
        Push(SB, r)  
While not stackEmpty(SB):  
    R = pop(SB)  
    If r != k:  
        Push(S, r)
```

Controllare che le parentesi, in una stringa, sono tante aperte quanto chiuse e messe correttamente

Check(S):

```
I = 1  
Err = false  
While not(err) and i <= len(S):  
    If s[i] == '(' or '[':  
        Push(P, s[i])  
    Else if s[i] == ')':  
        If stackEmpty(P) or top(P) != '[':  
            Err = true  
    Else
```

```
        Pop(P)
    Else if s[i] == ']':
        If stackEmpty(P) or top(P) != '[':
            Err = true
        Else
            Pop(P)
    i++
If not stackEmpty(P):
    Err = true
Return err
```