- algo iterativo
- due stack P_1 e P_2 lunghezza n di char
- un carattere CAR

mettere in stack P_3 char di P_1 (prima) e P_2 != CAR

```
algo ( P1 , P2 , P3 , CAR )
    c  boolean  turnoP1=true
c·tw while( !stackEmpty(P1) OR !stackEmpty(P2)) {
      c·tw if( !stackEmpty(P1) AND(turnoP1 OR stackEmpty(P2)){
            c·tif1 x=P1.pop
            c·tif1 if(x ≠ CAR)
              2c·tif2 P3.push(x)
                      turnoP1=false
          }
      c·tw if( !stackEmpty(P2) AND( !turnoP1 OR stackEmpty(P1)){
            c·tif3 y=P2.pop
            c·tif3 if(y ≠ CAR)
              2c·tif4 P3.push(y)
                      turnoP1= true
          }
    }
```

$T(n) = c + 3c\,tw + 2c \cdot tif_1 + 2c \cdot tif_3 + 2c\,tif_3 + 2c \cdot tif_1$

caso migliore

ho solo CAR in P_1 e P_2

$tif_2 = 0$ $tif_4 = 0$

$tif_1 = n$ $tif_3 = n$ $tw = 2n$

$T(n) = c + 3c \cdot 2n + 2cn + 2cn = c + 10cn = \Omega(n)$

caso peggiore

ho solo char ≠ CAR $tw = n$

$tif_2 = n$ $tif_4 = n$

$tif_1 = n$ $tif_3 = n$

$T(n) = c + 3cn + 2cn + 2cn + 2cn + 2cn$

$11cn + c = O(n)$

$T(n) = \Theta(n)$

BAAC → CZBYX

XYZA

1) turnoPy — pop C ≠ CAR — BAA / XYZA — C

2) ! turnoPy — pop A = CAR — BAA / XYZ — C

3) !turnoPy — pop Z ≠ CAR — BAA / XY — CZ

4) turnoPy — pop A = CAR — BA / XY — CZ

5) turnoPy — pop A — B / XY — CZ

6) turnoPy — pop B — ... / XY — CZB

7) ! turno Py — stackEmpty Py — .. / X — CZBY

pop Y ≠ CAR

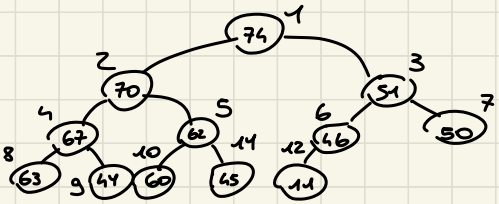
8) turnoPy — stackEmpty Py — ... / ... — CZBYX

pop X ≠ CAR

entrambi vuoti esco dal ciclo

2)

1	2	3	4	5	6	7	8	9	10	11	12
11	44	51	67	45	46	50	63	70	60	62	74



comincio dall' indice 6 perché é n/2

6) scambio 46 e 74

11, 44, 51, 67, 45, 74, 50, 63, 70, 60, 62, 46

5) passo a indice 5, scambio 45 e 62

11, 44, 51, 67, 62, 74, 50, 63, 70, 60, 45, 46

4) passo a indice 4

scambio 67 e 70

11, 44, 51, 70, 62, 74, 50, 63, 67, 60, 45, 46

3) passo a indice 3

scambio 51 e 74

11, 44, 74, 70, 62, 51, 50, 63, 67, 60, 45, 46

2) passo a indice 2

scambio 44 e 70

44, 70, 74, 44, 62, 51, 50, 63, 67, 60, 45, 46

scambio 44 e 67

44, 70, 74, 67, 62, 51, 50, 63, 44, 60, 45, 46

1) passo a indice 1

scambio 44 e 74

74, 70, 44, 67, 62, 51, 50, 63, 44, 60, 45, 46

scambio 44 e 51

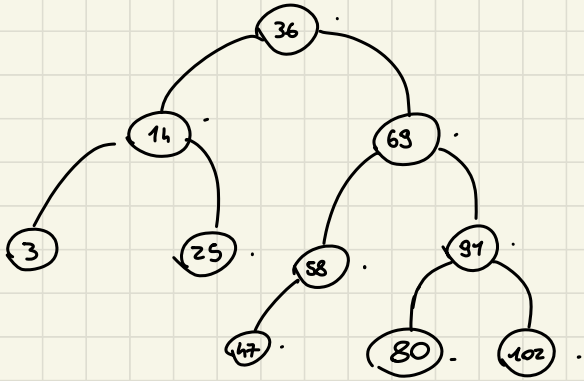
74, 70, 51, 67, 67, 44, 50, 63, 44, 60, 45, 46

scambio 44 e 46

74, 70, 51, 67, 67, 46, 50, 63, 44, 60, 45, 44

⇑ questo é un heap

3, 14, 25, 36, 47, 58, 69, 80, 91, 102
x x x x x x x



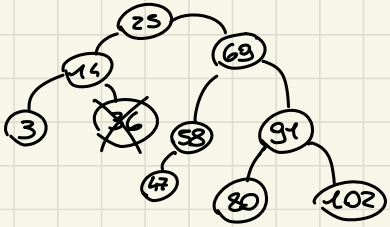
Anticipato
36, 14, 3, 25, 69, 58, 47, 91, 80, 102

simmetrico
3, 14, 25, 36, 47, 58, 69, 80, 91, 102

posticipato
3, 25, 14, 47, 58, 80, 102, 91, 69, 36

1) cancello 36

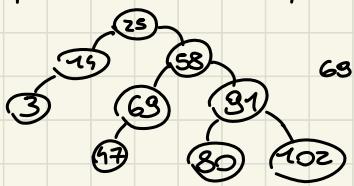
per cancellare 36 devo scambiarlo con il suo predecessore e poi cancellarlo



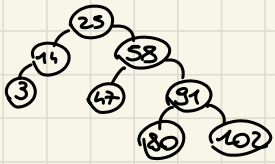
non ha figli
quindi lo cancello direttamente

2) cancello 69

come prima, scambio col predecessore e cancello

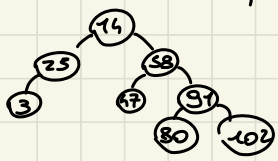


69 ora ha solo un figlio sinistro
metto come figlio sinistro di 58
47

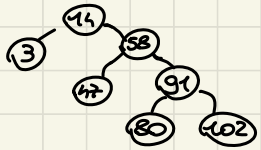


3) cancello 25

scambio 25 con il predecessore



25 ha un solo figlio sinistro, metto come figlio sinistro di 14 3



```
func (Q1, Q2, Q3, car){
    bodean turnoQ1 = true
    while (!QueueEmpty(Q1) || !QueueEmpty(Q2)){
        if(!QueueEmpty(Q1)) AND ( turnoQ1 || QueueEmpty(Q2)){
            x= deque (Q1)
            if(x ≠ CAR)
                Q3.enque(x)
            turnoQ1= false
        }
        if(!QueueEmpty(Q2 )) AND (!turnoQ1 || QueueEmpty(Q1)){
            y= deque(Q2)
            if(y ≠ CAR)
                Q3.enque(y)
            turnoQ1= true
        }
```

pila p ordine decrescente

7 ins 6 e 4

5
2
1

<6 allora push

e

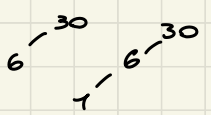
5
2
1

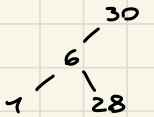
>6

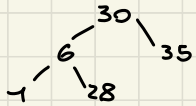
7

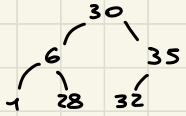
```
if (x ≥ p.top)
    p.push(x)
if(y≥ p.top)
    p.push(y)

while(!stackEmpty(p))   {
      app=p pop
      pz.push(app)
      if(p.top< x )
          ap push(x)
      if(p.top<y)
            push(y)


}
```

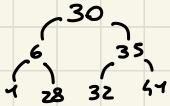
30



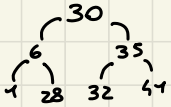






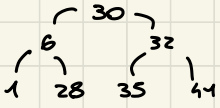


devo cancellare 35 da



avendo 2 figli

devo prendere il predecessore e scambiarlo con 35



adesso devo cancellare 35, che non avendo nessun figlio mi permette di mettere il puntatore sinistro del padre a null