

# SKLAD

- `potisni(x)` : doda  $x$  na vrh
- `poči()` : izbrise element iz vrha in ga vrne
- `dolžina()` : vrne število elementov na skladu
- `pokukaj()` : vrne zgornji element

Zahtevamo, da so vse operacije  $O(1^*)$ .

---

Implementiraj strukturo `MinSklad`, ki ima iste operacije kot sklad, pa še metodo `min`, ki vrne minimalen element na skladu. Vse operacije morajo delati v  $O(1)$ .

Ideja: Ustvarimo pomožni sklad, ki shranjuje minime.

```
class MinSklad:
```

```
    sklad = sklad()
    mini = sklad()
```

```
    def potisni(x):
        sklad.potisni(x)
```

```
if len(mini) == 0:  
    mini.potisni(x)  
elif mini.pokukaj() >= x:  
    mini.potisni(x)
```

```
def poči():  
    if sklad.pokukaj() == mini.pokukaj():  
        mini.poči()
```

```
    return sklad.poči()
```

```
def pokukaj():  
    return sklad.pokukaj()
```

```
def min():  
    return mini.pokukaj()
```

Trik: Brez pomožnega sklada:

```
class MinSklad:
```

```
    sklad = sklad()
```

```
    min = ∞
```

```
    def potisni(x):  
        if x < min:  
            sklad.potisni(2 * x - min)  
            min = x  
        else:  
            sklad.potisni(x)
```

```
    def poči():  
        if min > sklad.pokukaj():  
            y = sklad.poči()
```

$x = \min$

$\min = 2 \cdot x - y$

return  $x$

else:

return sklad.poci()