

# Obilig 1

Brage Wiseth

September 7, 2022

## Oppgave 2

For å implementere "Teque" bruker jeg to sirkulære arrays **f** og **b** som representerer hver sin halvdel av køen. Hver array har en tilhørende **front** og en **back** peker. De døpes **f\_front**, **f\_back**, **b\_front**, **b\_back**. **f** for første halvdel **b** for andre halvdel. **b\_front** fortsetter køen der **f\_back** slapp. Det vil si at hvis **b\_front** tilsvare indeks 5 så vil **f\_back** tilsvare indeks 4. **f** og **b** har hver sin referanse til mengder elementer i seg selv. Det brukes til å balansere de to.

### Algorithm: Push front

**Input:** Ett element **x**

**Output:** Plasserer **x** fremst i køen

```
1 procedure push_front(x)
2   f[f_front] ← x
3   f_front++ and f_size++
4   if f_front > |f|-1 then
5     f_front ← 0
6   balanser()
```

### Algorithm: Push back

**Input:** Ett element **x**

**Output:** Plasserer **x** sist i køen

```
1 procedure push_back(x)
2   b[b_back] ← x
3   b_back-- and b_size++
4   if b_back < 0 then
5     b_back ← |b|-1
6   balanser()
```

### Algorithm: push middle

**Input:** Ett element **x**

**Output:** Plasserer **x** midt i køen

```
1 procedure push_middle(x)
2   if f_size < b_size then
3     f[f_back] ← b[b_front-1]
4     b[b_front-1] ← x
5     f_back-- and f_size++
6     if f_back < 0 then
7       f_back ← |f|-1
8   else then
9     b[b_front] ← x
10    b_front++ and b_size++
11    if b_front > |b|-1 then
12      b_front ← 0
```

### Algorithm: get

**Input:** En indeks **i**

**Output:** element på indeks **i**

```
1 procedure get(i)
2   if i < f_size then
3     j ← (f_front-1) - i
4     if j < 0 then
5       return f[|f| + j]
6     else return f[j]
7   else then
8     j ← b_(front-1) - (i-f_size)
9     if j < 0 then
10      return b[|b| + j]
11    else return b[j]
```

Til slutt har vi hjelpemetoden **balanser()**:

**Algorithm: balanser**

```
1 procedure balanser()
2   d  $\leftarrow$  f_size - b_size
3   if d > 1 then
4     f_back++
5     b[b_front]  $\leftarrow$  f[f_back]
6     f[f_back]  $\leftarrow$  null
7     b_front++
8     if b_front > |b|-1 then
9       b_front  $\leftarrow$  0
10    b_size++ and f_size--
11  if d < -1 then
12    b_front--
13    f[f_back]  $\leftarrow$  b[b_front]
14    b[b_front]  $\leftarrow$  null
15    f_back--
16    if f_back < 0 then
17      f_back  $\leftarrow$  |f|-1
18    b_size-- and f_size++
```