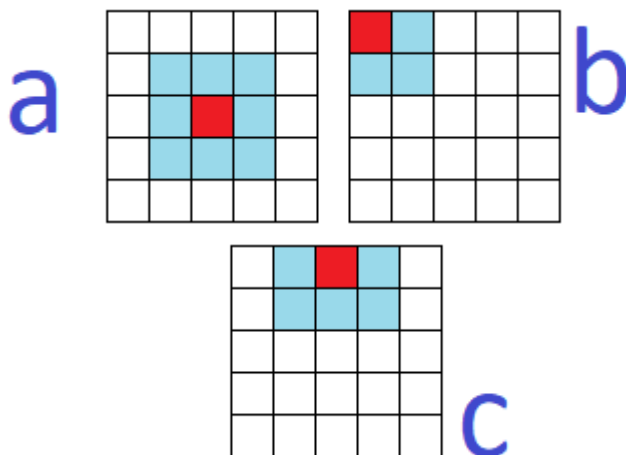


Litt om tankegangen rundt todimensjonale lister

Når vi skal sjekke hvilke elementer som er naboer til et element i ei todimensjonal liste er det viktig å tenke på når vi kan få problemer. Bilde a) under viser de 8 naboene til et element. Problemene dukker opp når elementet vi skal sjekke naboene til ligger langs en kant eller i et hjørne. Se bilde b) og c) under. Å være utenfor kanten vil si at indeksen til naboene vi vil legge til er lavere enn den laveste indeksen (0), eller høyere eller lik lengden på den ytterste lista (rader), eller de indre listene (kolonner).



Husk at de todimensjonale listene vi har i IN1000 har følgende struktur, her illustrert med firkanter.

```

[['■', '■', '■', '■', '■'],
 ['■', '■', '■', '■', '■'],
 ['■', '■', '■', '■', '■'],
 ['■', '■', '■', '■', '■'],
 ['■', '■', '■', '■', '■']]
    
```

Den første indeksen vi gir for å få tak i et element bestemmer altså hvilken rad vi skal ha tak i. For eksempel gir indeks [3] den fjerde raden. Hvis vi skal ha tak i det andre elementet i den fjerde raden, skriver vi [3][1].

Todimensjonale lister og for-løkker

Som vi har sett på i timen hører todimensjonale lister og doble for-løkker sammen. Både når vi oppretter og når vi går gjennom todimensjonale lister, bruker vi doble for-løkker.

Det er blant annet to måter å opprette todimensjonale lister på. Begge starter med ei tom liste, og vi antar at vi har et heltall (int) *rader* som er antall rader vi skal ha, og et heltall *kolonner*, som er antall kolonner, altså elementer i de indre listene:

```

rader = 5
kolonner = 5
todimensjonalListe = []
    
```

A)

```
for i in range(rader):
    indreListe = []
    for j in range(kolonner):
        indreListe.append('■')
    todimensjonalListe.append(indreListe)
```

Det som skjer her er at for hver rad vi vil ha, lager vi ei tom liste. Så, for hvert element vi vil ha i lista, altså kolonne, legger vi til det elementet vi vil ha. Så må vi legge til den indre lista i hovedlista vår.

B)

```
for i in range(rader):
    todimensjonalListe.append([])
    for j in range(kolonner):
        todimensjonalListe[i].append('■')
```

Her legger vi ei ny liste rett inn i hovedlista. Siden vi legger til ei ny liste for hver iterasjon, vet vi at det er éi liste når vi skal legge inn noe første gangen, det vil si indeks 0, og derfor kan vi bruke indeksen *i* til å få tilgang til lista vi har oppretta, og vi kan legge elementene rett inn i den. Denne måten å gjøre det på er kortere.

Begge metodene lager helt identiske todimensjonale lister:

```
[['■', '■', '■', '■', '■']]
['■', '■', '■', '■', '■']
['■', '■', '■', '■', '■']
['■', '■', '■', '■', '■']
['■', '■', '■', '■', '■']
```