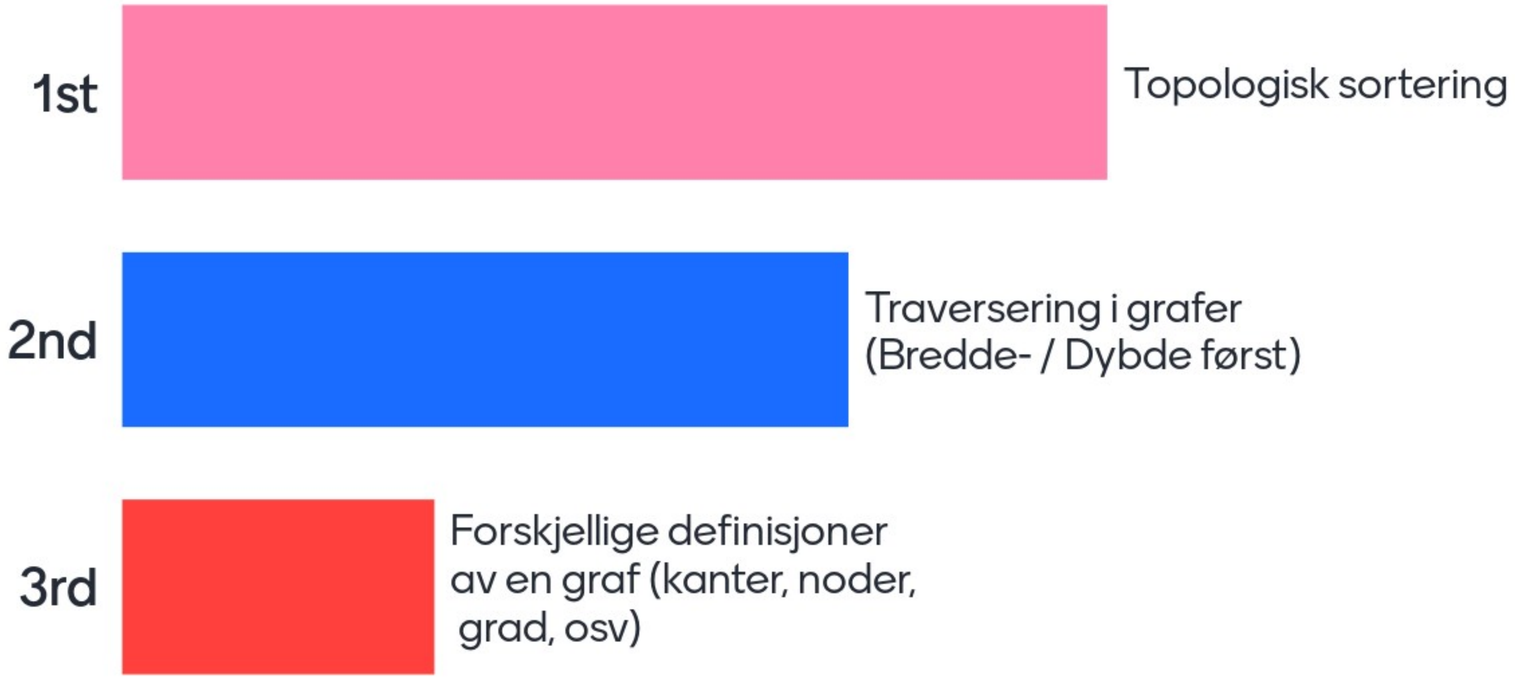
Velkommen til IN2010 gruppe 6

Dagens plan

- Gjennomgang av pensum
- Sette opp en graf (live koding)
- Lab

Hva var vanskeligst med pensum denne uken?



Hva er en graf?

- Er noe som er bygget opp av noder og kanter
  - En Node er ***noe*** som holder på data
  - En Kant er ***noe*** som beskriver koblingen mellom to noder
- Grafen ***G*** består av en mengde noder ***V*** og en mengde kanter ***E***

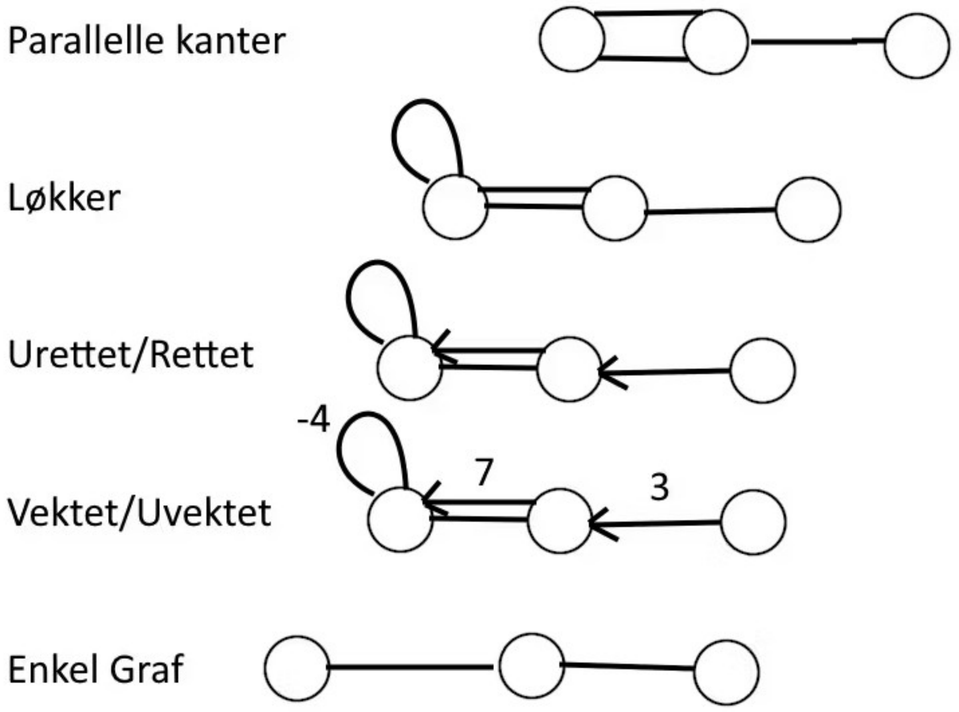
- Eksempel: Jorda er en graf, V: land, E: landegrenser
  - Norge, Sverige og Danmark er noder
  - Det er en kant mellom Norge og Sverige
  - Det er ikke en kant for Norge og Danmark

Litt mer om grafer

- Her er noen betegnelser som beskriver en graf:

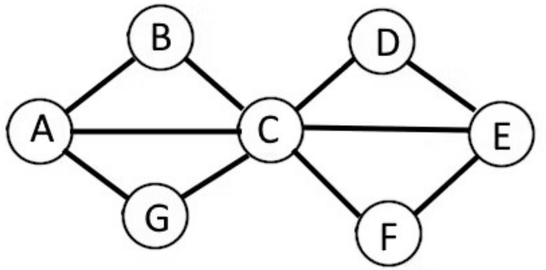
- En kant kan bli representert på forskjellige måter:
  - En urettet graf: {A, B}
  - En rettet graf: (A, B)
  - En urettet graf med rettede kanter: (A, B) og (B, A)

- En vekt sier noe om hvor mye det koster å traversere
- Det kan bli representert med en ny mengde ***W*** som inneholder følgende verdier:
  - (A, B, 3)
  - (B, C, 7)
  - (C, C, -4)



Enda mer om grafer

- Generelt når man traverserer er det kantene som bestemmer hvor vi kan gå
- Noen nye definisjoner: Veier og stier
  - En ***sti*** er en traversering av grafen der hvor ingen noder gjentas
  - En ***vei*** er en traversing av grafen der hvor ingen kanter gjentas



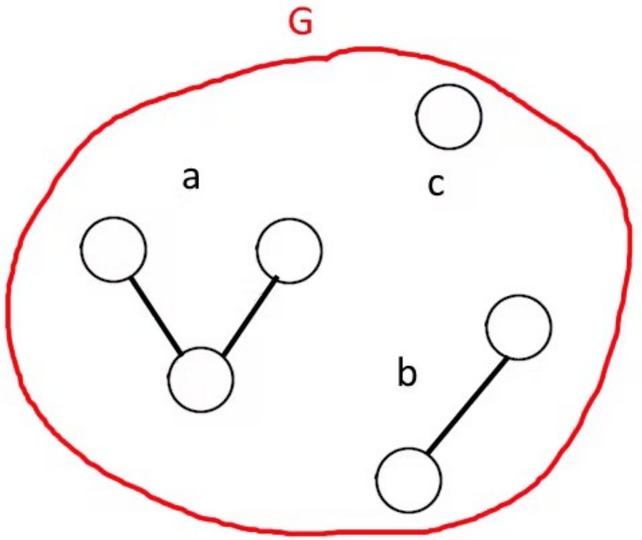
sti: A > B > C > D > E > F

vei: A > B > C > D > E > F > C > G > A >C > E

Enda enda mer om grafer

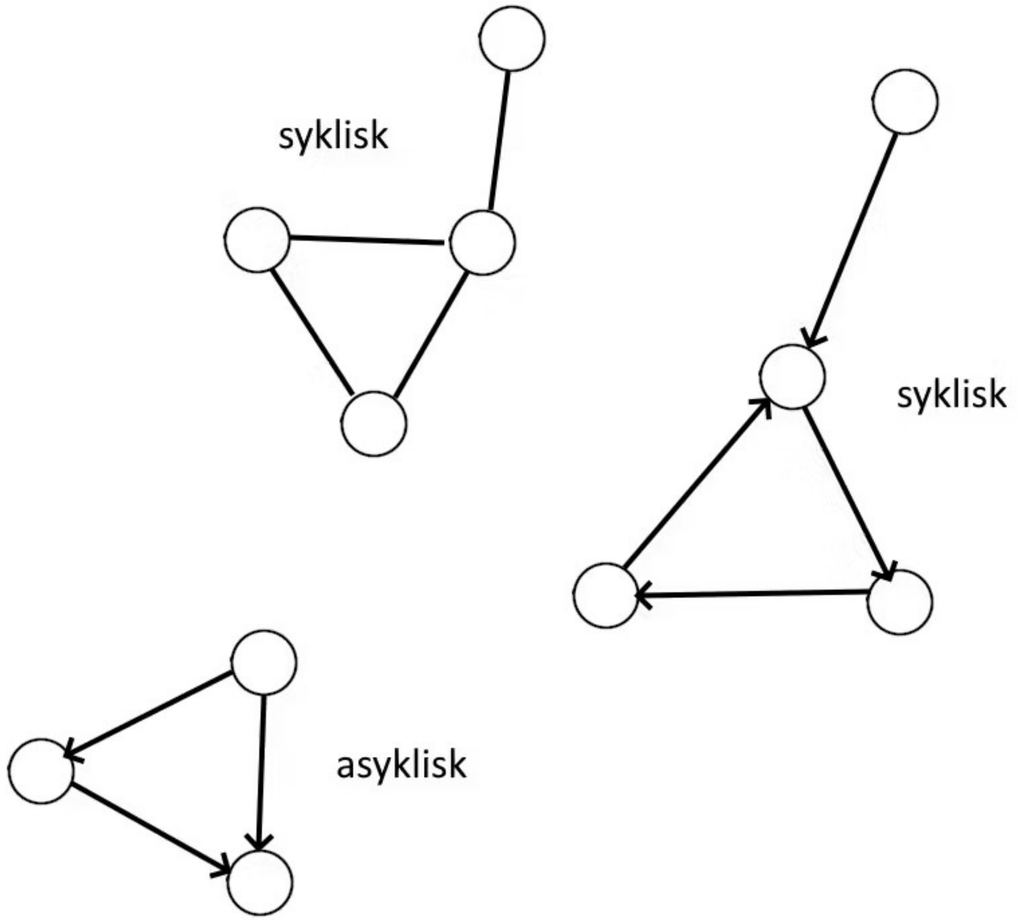
- Sammenhengende grafer er når:
  - Det finnes en **sti** med alle nodene i **V**
- Ikke sammenhengende grafer består av komponenter

- Til høyre er det en graf ***G*** med komponent a, b og c



Enda enda enda mer om grafer

- Sykler i en graf er når det finnes en sti der hvor start og slutt noden er den samme
  - Med andre så er det mulighet for å gå rundt i sirkel inne i en graf
- En graf som er verdt å kjenne til er DAG, Directed asyclic graph
  - En graf som er båre rettet og uten sykler



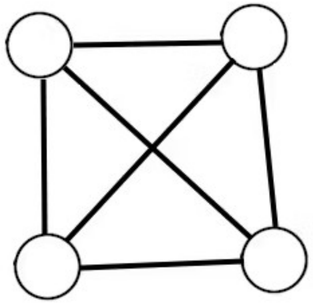
Siste om grafer

- En komplett graf er når alle noder har en kanter til alle andre noder

- Størrelse til en graf er bestemt av antall noder og antall kanter
  - |E| = (|V| * |V| - |V|)/2
  - 6 = (4 * 4 - 4) / 2 = 12 / 2

- Annerledes i store O
  - O((|V|*|V|-|V|)/2) => O(|V|^2) (velger største ledd)
  - Så O(|E|) = O(|V|^2)

- En **tett graf** er en graf med mange kanter
- En **tynn graf** er en graf med få kanter



Over til traversering

- Generelt har man en liste som holder styr på hvilke noder man skal besøke
- Denne listen blir fylt opp med nabo noder på den noden man ligger på
- Denne uka er det fokus på BFS (bredde først) og DFS (dybde først)

Bredde vs Dybde

- Bredde først bruker en kø
- Dybde først bruker en stack
- Det vil si at i Bredde først så fokuserer vi på det første elementet i listen (kø)
- Derimot i Dybde først så fokuserer vi på det siste elementet i listen (stack)

```python
def DybdeFørstSøk(G, s):
    V, E, w = G
    visited = set()
    stack = [s]
    result = []

    while stack:
        v = stack.pop()
        if v not in visited:
            result.append(v)
            visited.add(v)
            for u in E[v]:
                stack.append(u)
    return result

def BreddeFørstSøk(G, s):
    V, E, w = G
    visited = set([s])
    queue = [s]
    result = []

    while queue:
        v = queue.pop(0)
        result.append(v)
        for u in E[v]:
            if u not in visited:
                visited.add(u)
                queue.append(u)
    return result
```

TOPSort

- En algoritme for å lage en sortert liste basert på en DAG
- Finner alle nodene med ingen ingang (start noder)
- Går til neste node fra start nodene
- Deretter fjerner kanten mellom noden man dro fra og til
- Sjekker om noden har ingen inganger og legger dem til i output

```
def TOPsort(G):
    V_1, E_1 = G
    V = V_1.copy()
    E = E_1.copy()

    stack = []
    output = []

    for v in V:
        if v not in E.keys():
            stack.append(v)

    while stack:
        v = stack.pop()
        output.append(v)
        for u in E[v]:
            E[v].remove(u)
            if u not in E.keys():
                stack.apend(u)

    if len(output) < len(V):
        print("Noe galt, sykel i grafen")
    return output
```

Over til live koding :)

- Demonstere land eksempelet
- Lage land og kontinenter