

# IN1010 Seminarartime - Uke 13

Rekursjon



# Agenda

- Litt om rekursjon
- Tips til å forstå rekursjon
- Livekoding
- Seminaroppgaver og oblig



# Rekursjon innebærer at...

Metoder kaller på seg selv

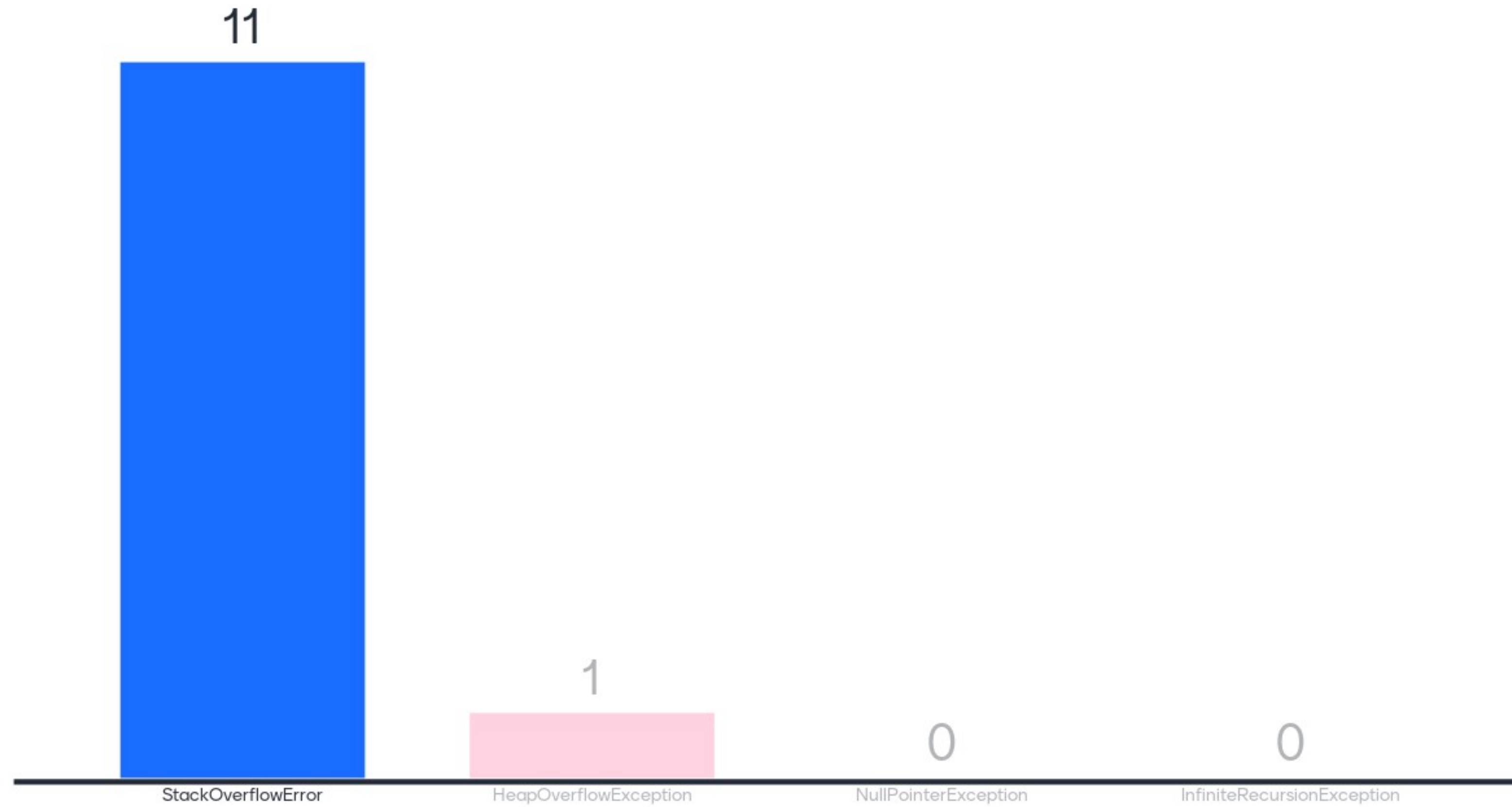


or

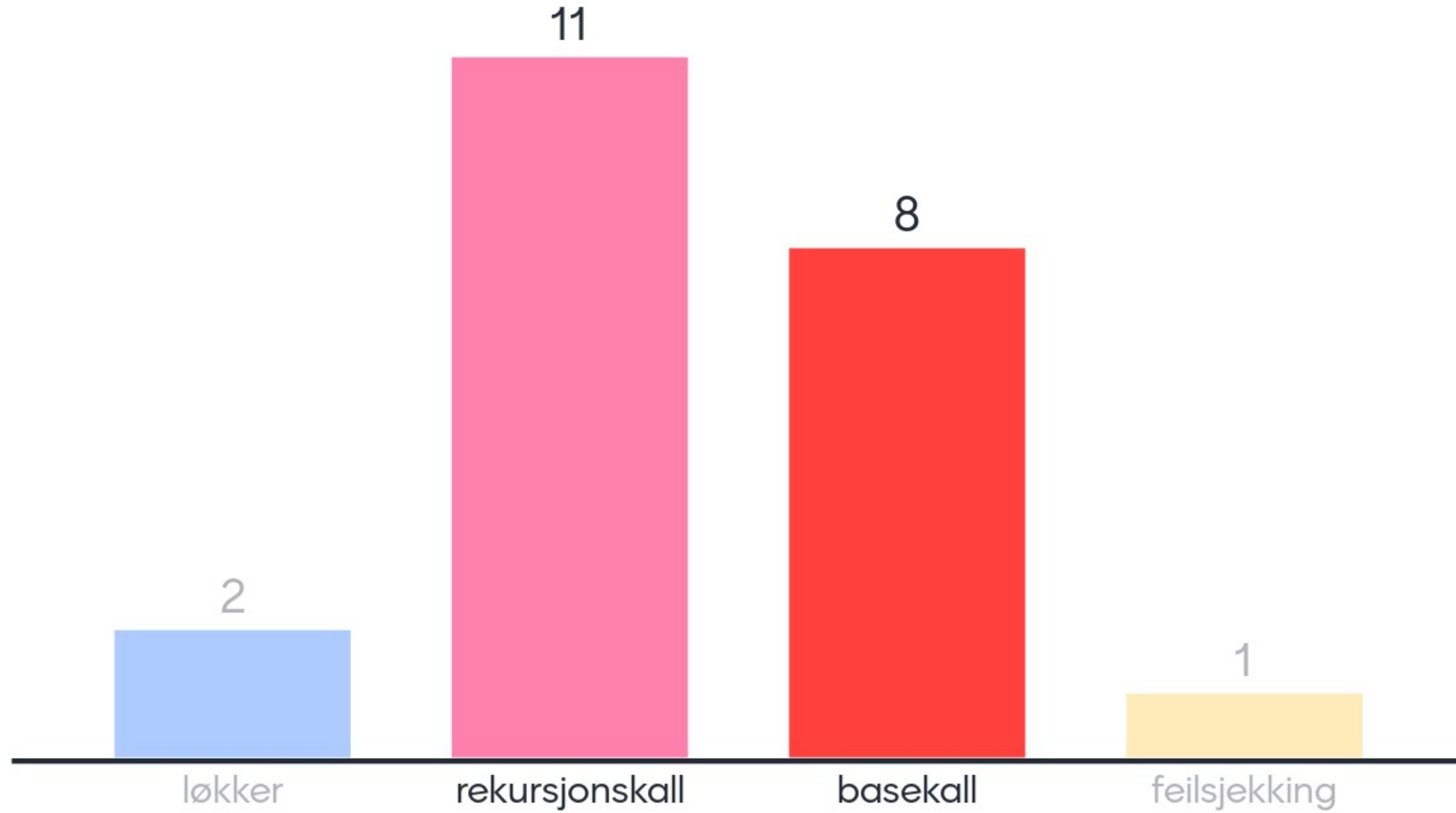
Metoder øker problemet helt til løsningen kommer



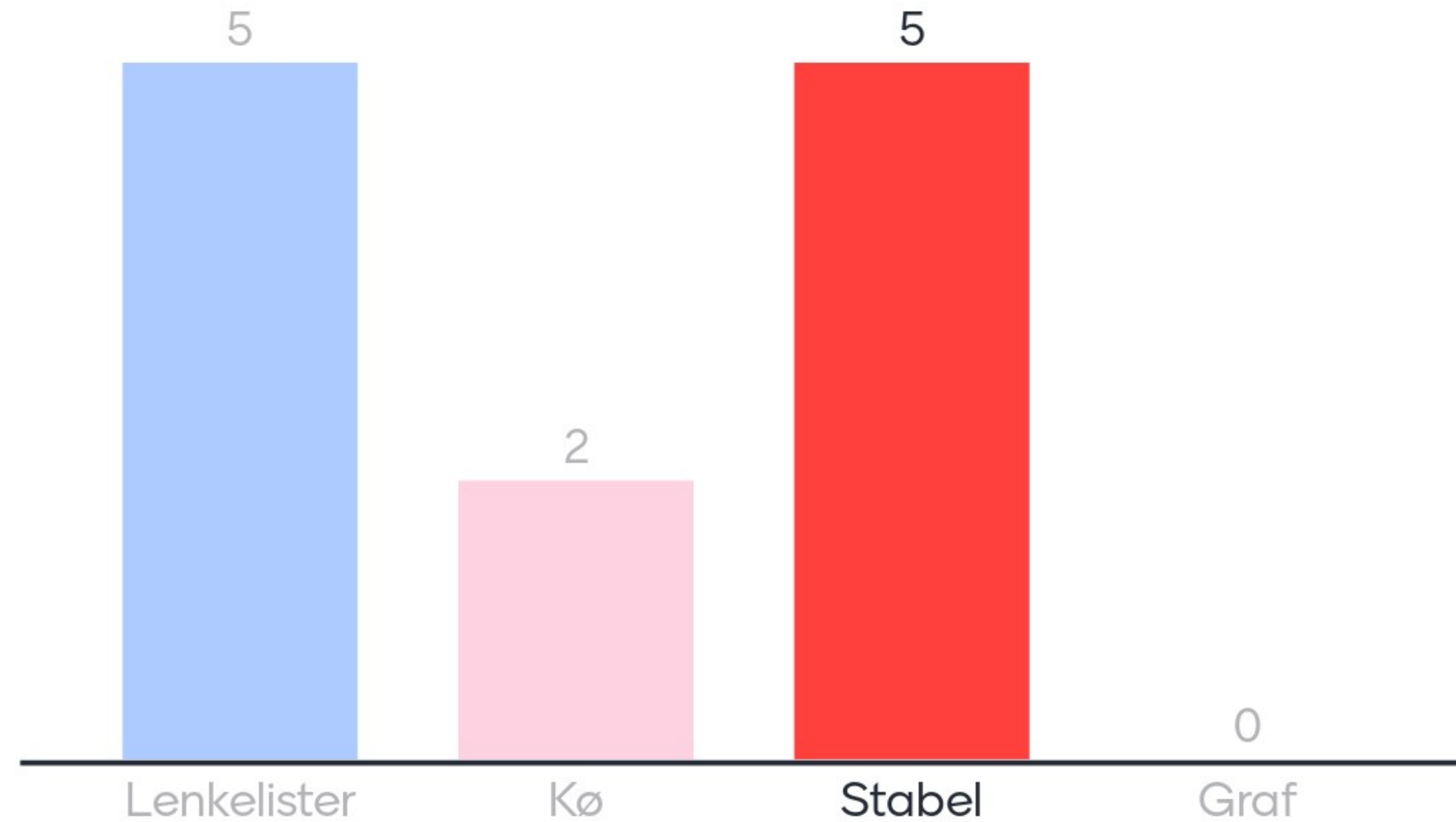
# Uendelige rekursive kall kan generere en..



# En rekursiv metode består av...

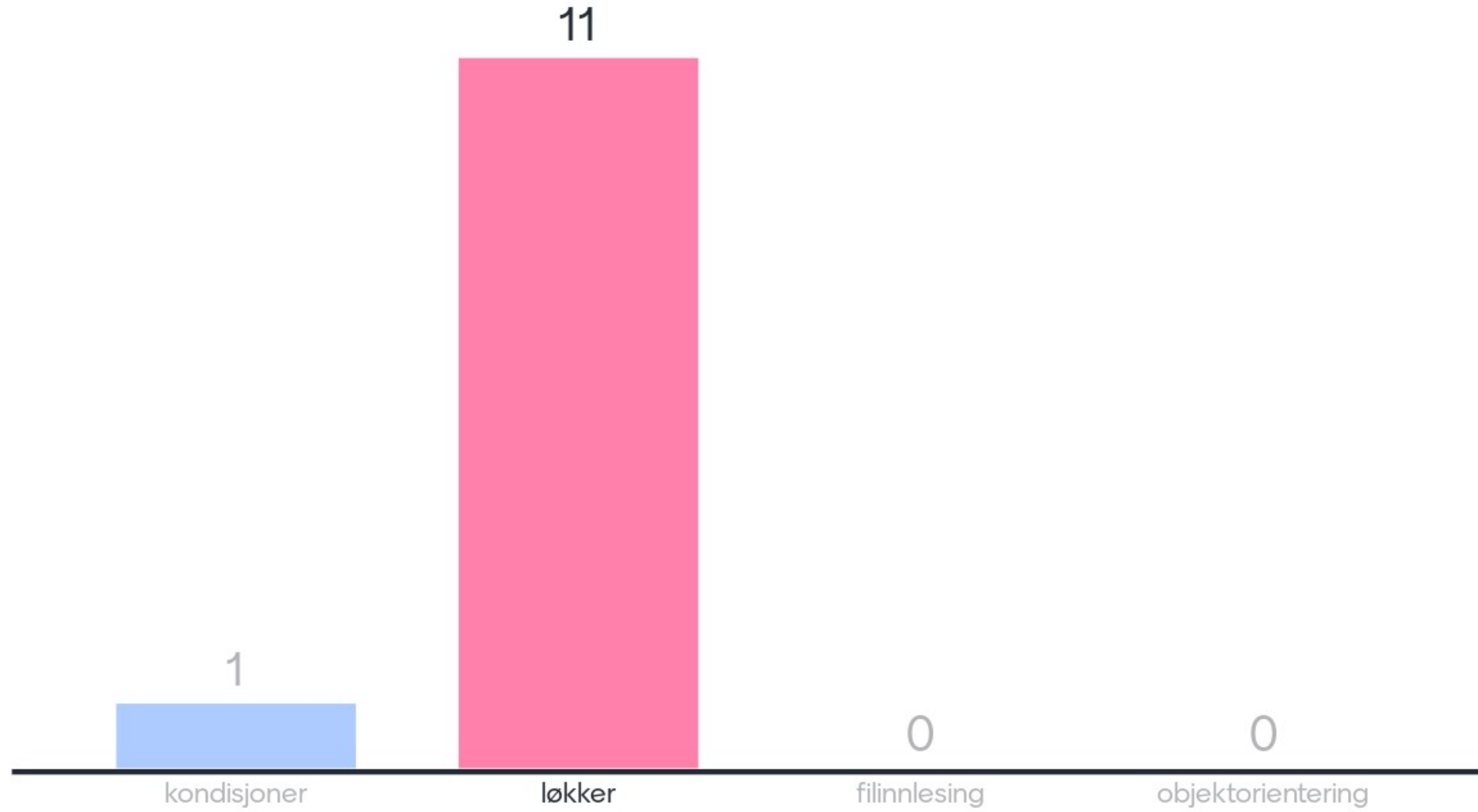


# Hvilken datastruktur er tett knyttet til rekursjon?





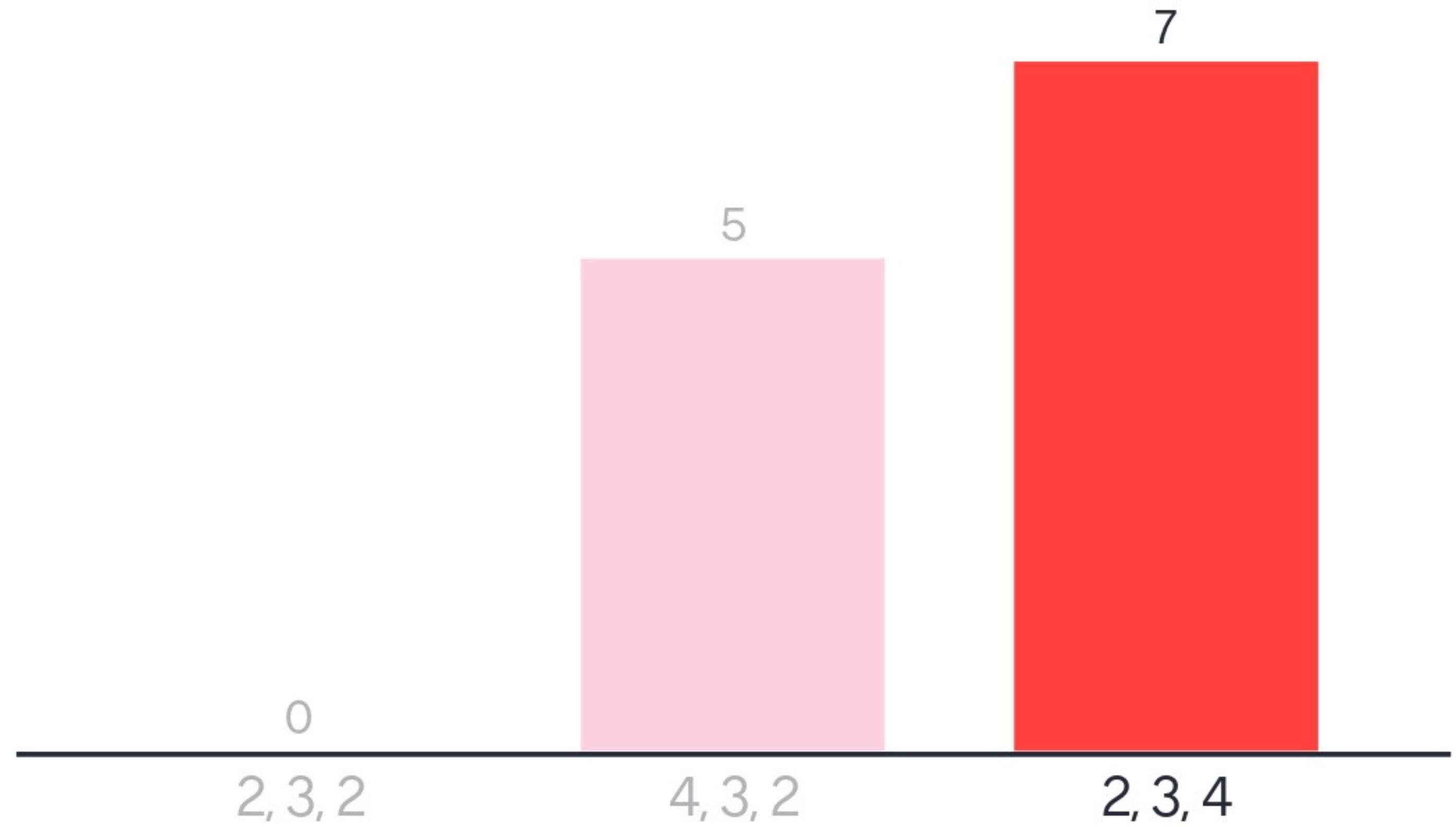
# Rekursjon ligner på...



# Hva blir skrevet ut her?

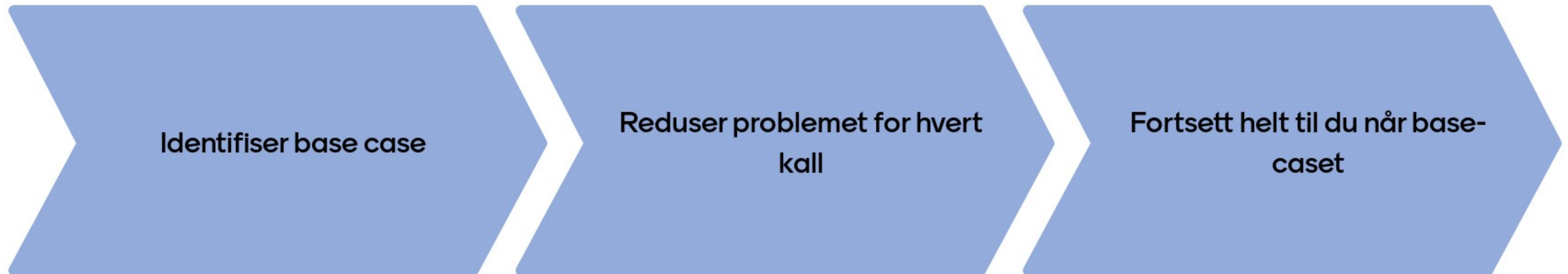
```
Run | Debug
public static void main(String[] args) {
    skrivRekursivt(4);
}

public static void skrivRekursivt(int tall) {
    if (tall < 2) {
        return;
    }
    skrivRekursivt(tall - 1);
    System.out.println(tall);
}
```





# Gangen i rekursjon



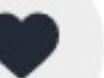
# Rekursjon

- En "elegant" måte å løse problemer på
- Består av et base-kall og rekursjonskall
- Mye av jobben handler om å identifisere et base-kall
- Viktig å redusere problemet for hvert kall (nærme seg base-kallet)



# Tips til å lære seg rekursjon

- Litt vrient å tenke rekursivt, hjelper å tegne
- Leke seg frem med å endre når ting printes og ikke
- Masse YT-videoer, eks. [denne](#)



# Livekoding

Reversere og skrive ut en streng ved hjelp av rekursjon



## Hva nå?

- [Seminaroppgaver](#)
- [Oblig](#)
- Lese seg opp, og utforske litt
- Enda en video om rekursjon: [her](#)
- Kjent tilnærming: vanlig metode som "setter i gang" rekursjonen



# Lenker

- Første video: <https://www.youtube.com/watch?v=ngCos392W4w>
- Andre video: <https://www.youtube.com/watch?v=B0NtAFf4bvU&t=18s>

