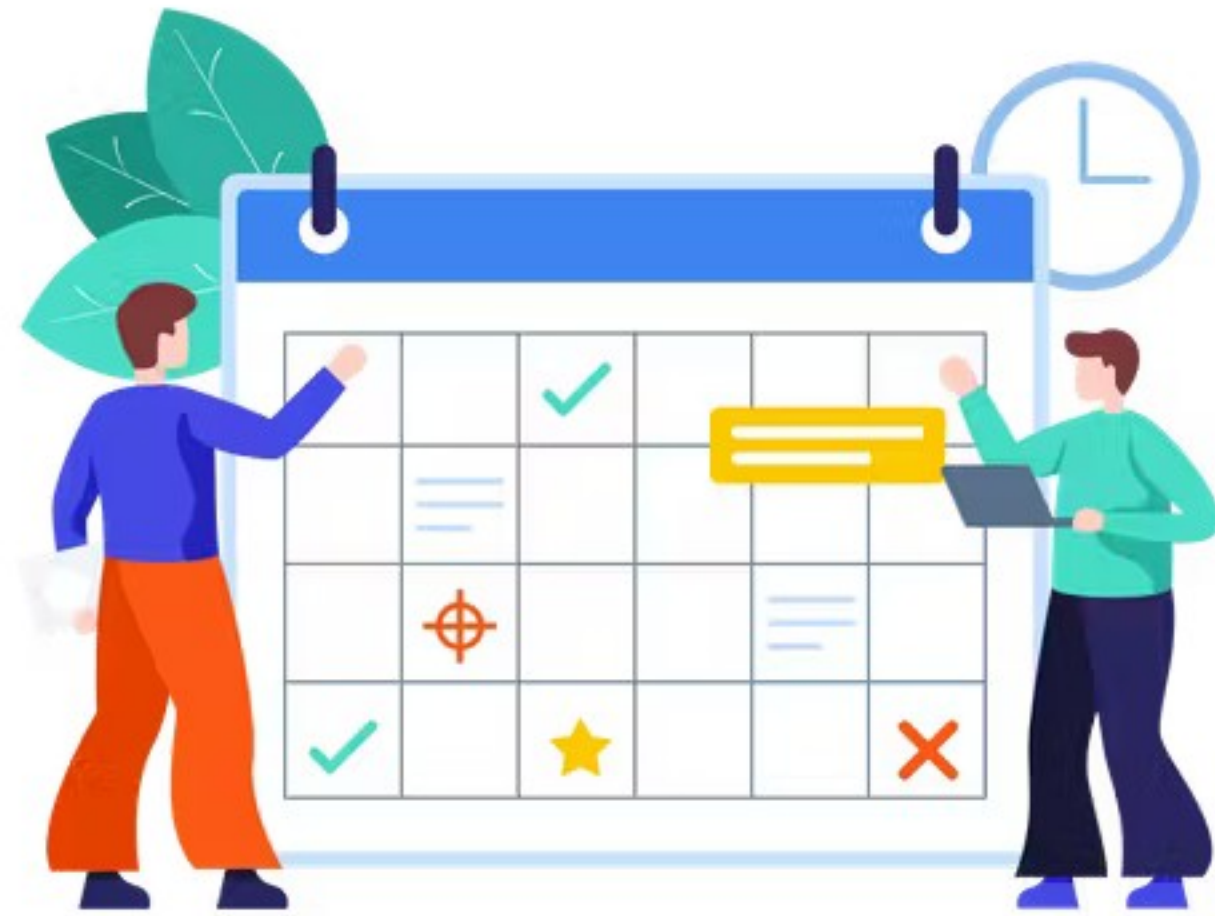


IN2010 Gruppe 4

Amadu Swaray

Bli med!



Dagens Plan

- Praktisk info
- Hvordan å jobbe med faget
- Kattis intro
- Pause
- Pensumgjennomgang
- Gruppe Oppgaver



Hvem er jeg?

- 5. Året prosa
- Trønder
- Musikk og Mat

Praktisk Info

- E-post: amadus@ifi.uio.no
- Forum: Astro Discourse
- 1 Obligatorisk Innlevering
- 3 Innleveringer
- Frist på Torsdager
- IN2010 ≠ IN1010 | IN1000



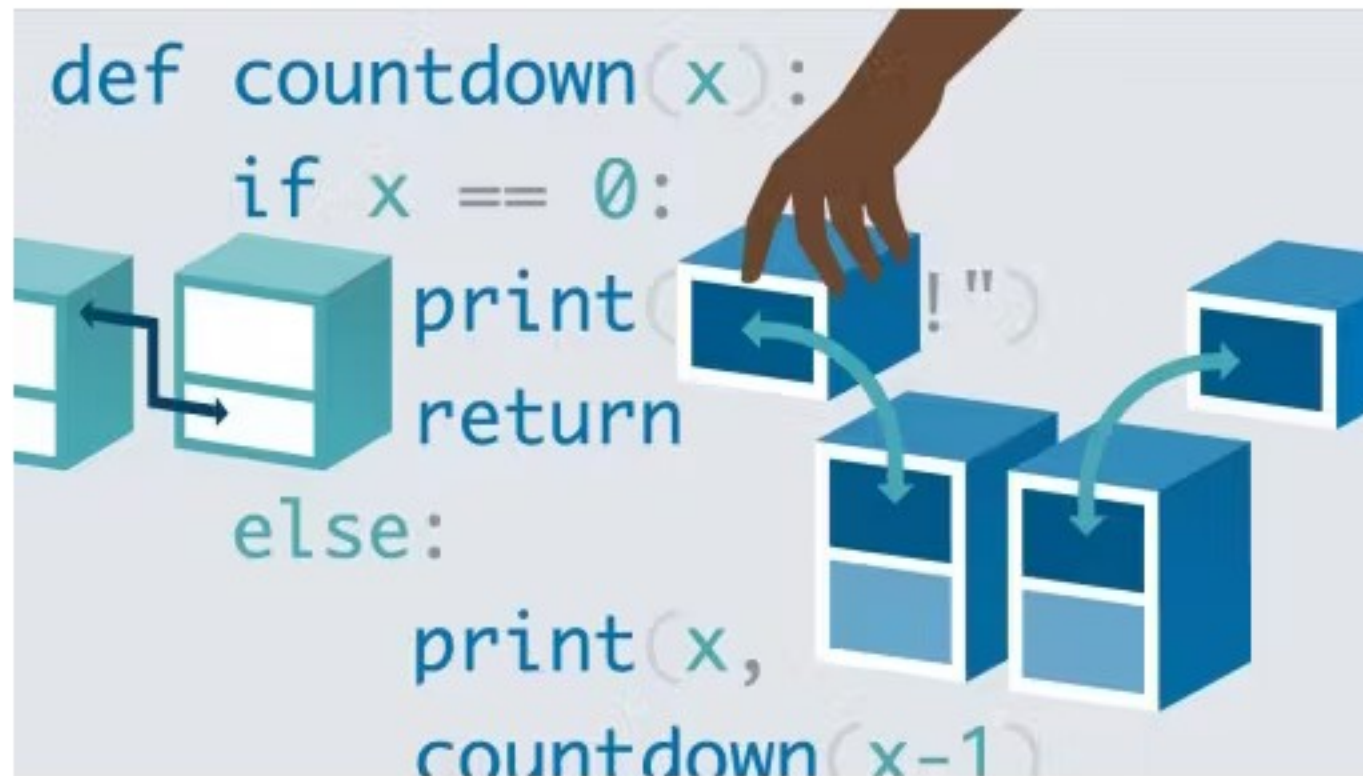
Infoside

<https://github.com/amaduswaray/IN2010-Gruppe-4/tree/main>

Hva skal vi gjøre i gruppetimene

1. Diskutere
2. Jobbe med oppgaver
3. Lære





IN2010 som emne

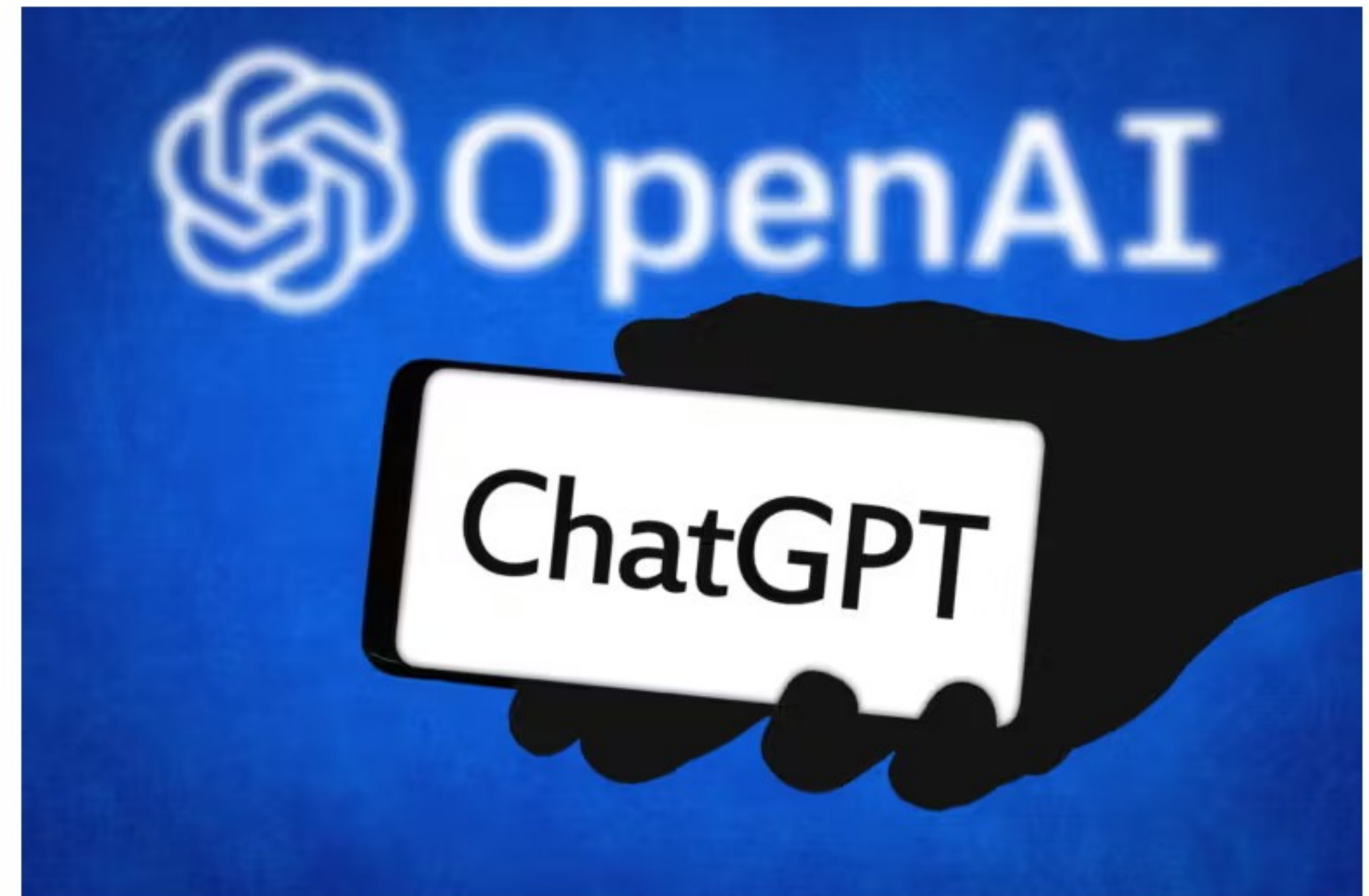
- Top 3 emne på IFI
- Gull for tekniske intervju
- Blir veldig god til å kode


```
def Kodetest(graph, root, target):  
  
    if target > len(graph) or root > len(graph):  
        return 'Input is outside the matrix'  
  
    stack = []  
    visited = []  
    stack.append(root)  
  
    while len(stack) > 0:  
        node = stack[-1]  
        visited.append(node)  
        foundNodes = False  
        for i in range(len(graph)):  
            if graph[node][i] == 1:  
                if i == target:  
                    visited.append(i)  
                    return {  
                        'visited': visited,  
                        'root': root,  
                        'target': target  
                    }  
                else:  
                    if i not in visited and i not in  
stack:  
                        foundNodes = True  
                        stack.append(i)  
                        break  
  
        if not foundNodes:  
            stack.pop()
```

Kodetest

Chat GPT

- Beste verktøyet
- Kan løse alle obliger



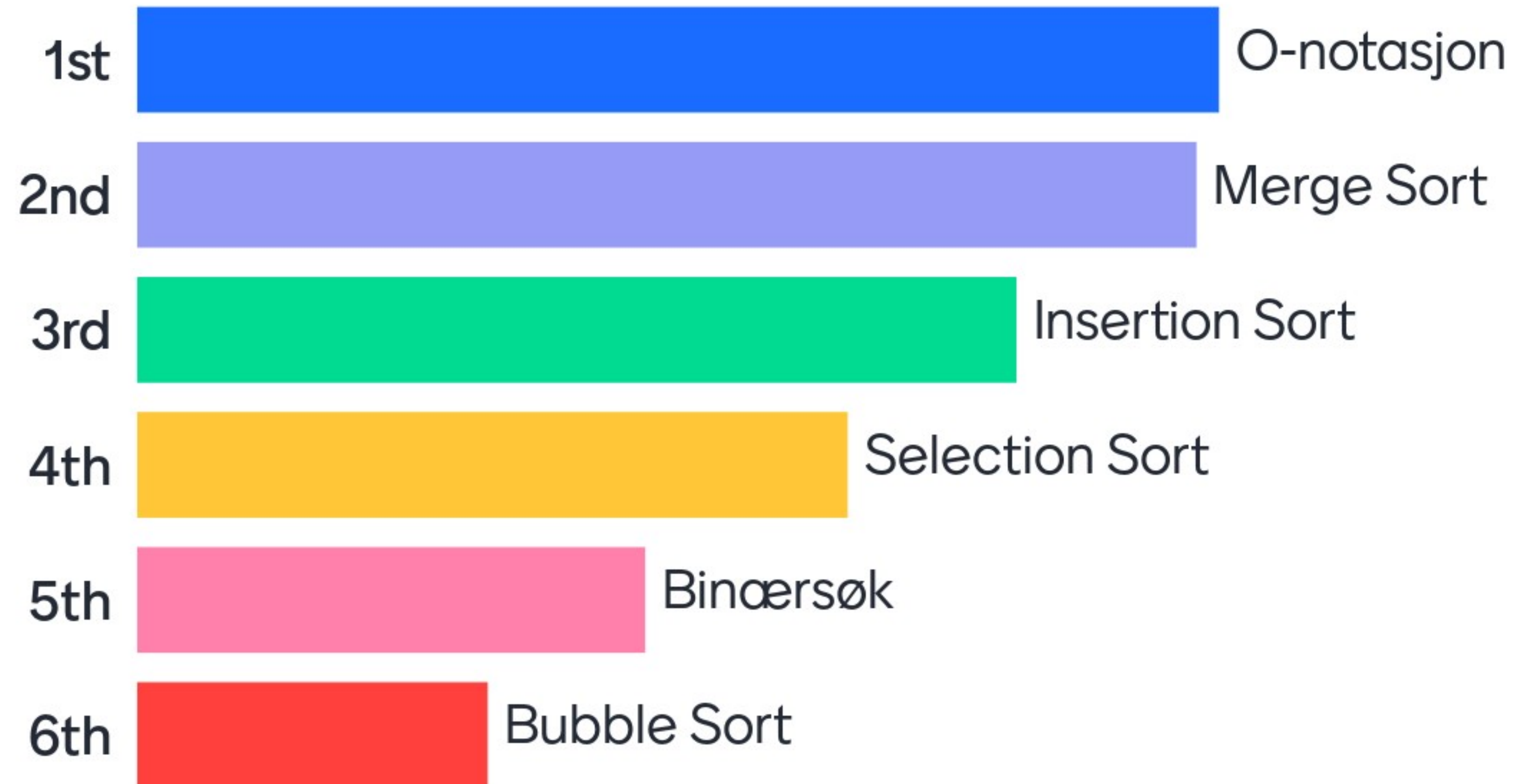
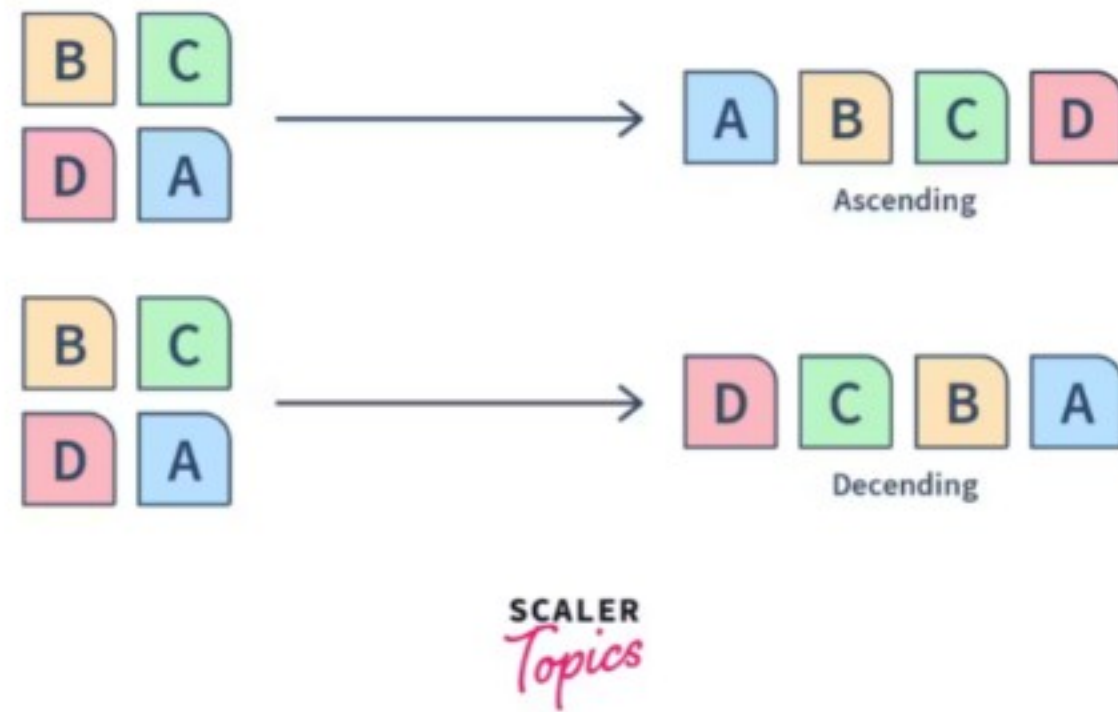


Kattis

Pause

Pensumgjennomgang

Rank Temaene fra denne og forrige ukes foelesning :)



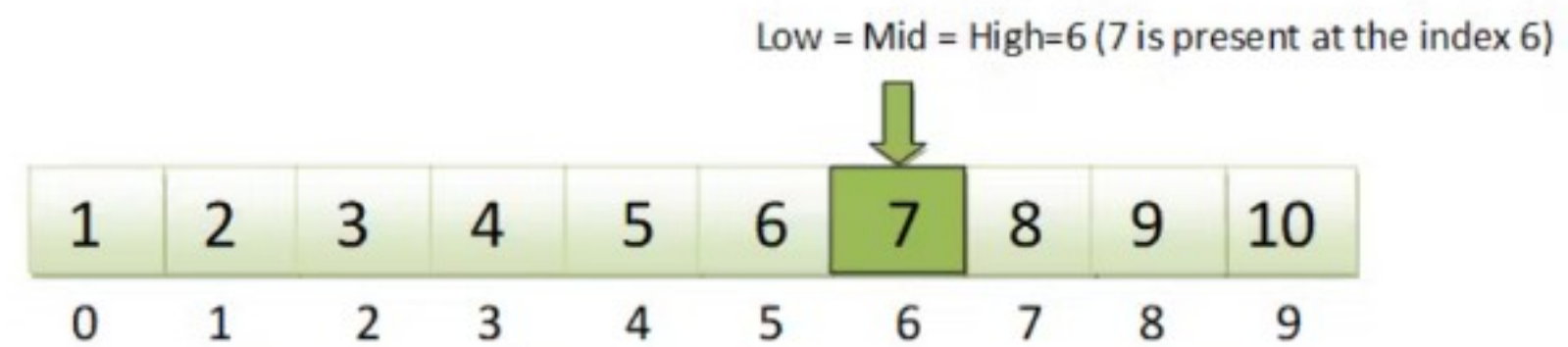
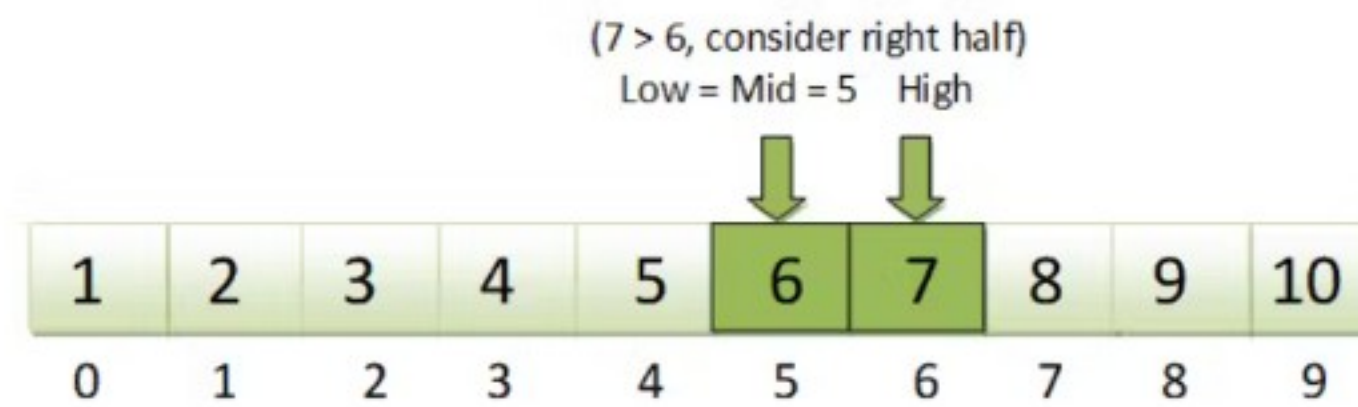
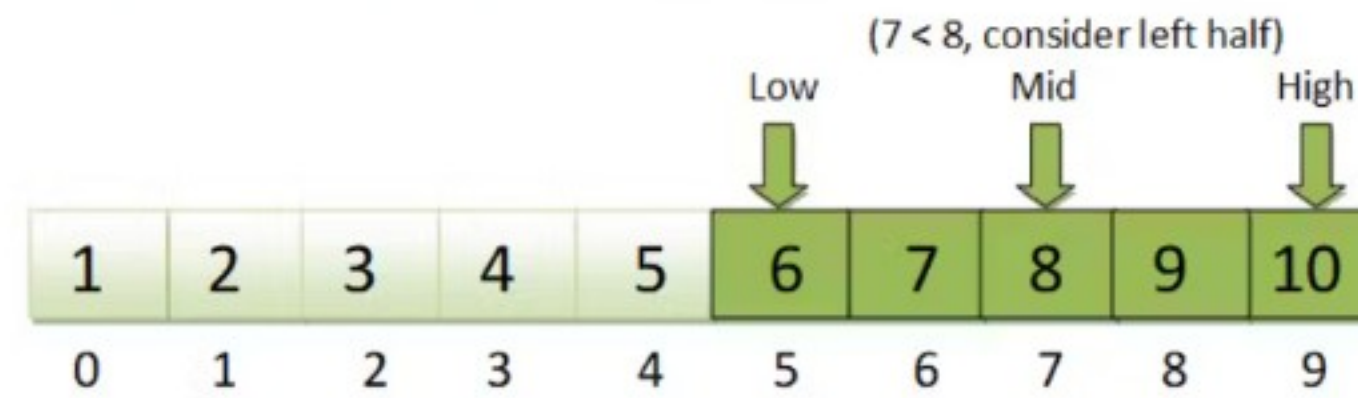
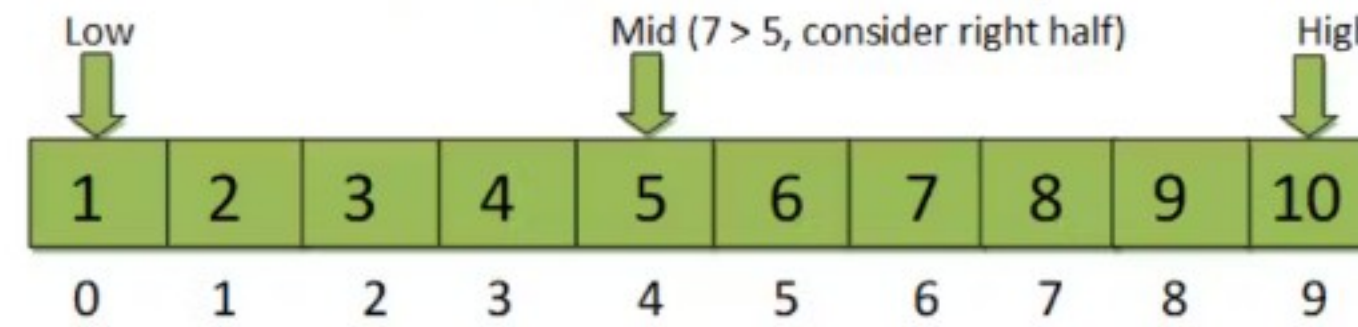
Bindærsøk

Ide: Slå opp ord i en ordbok



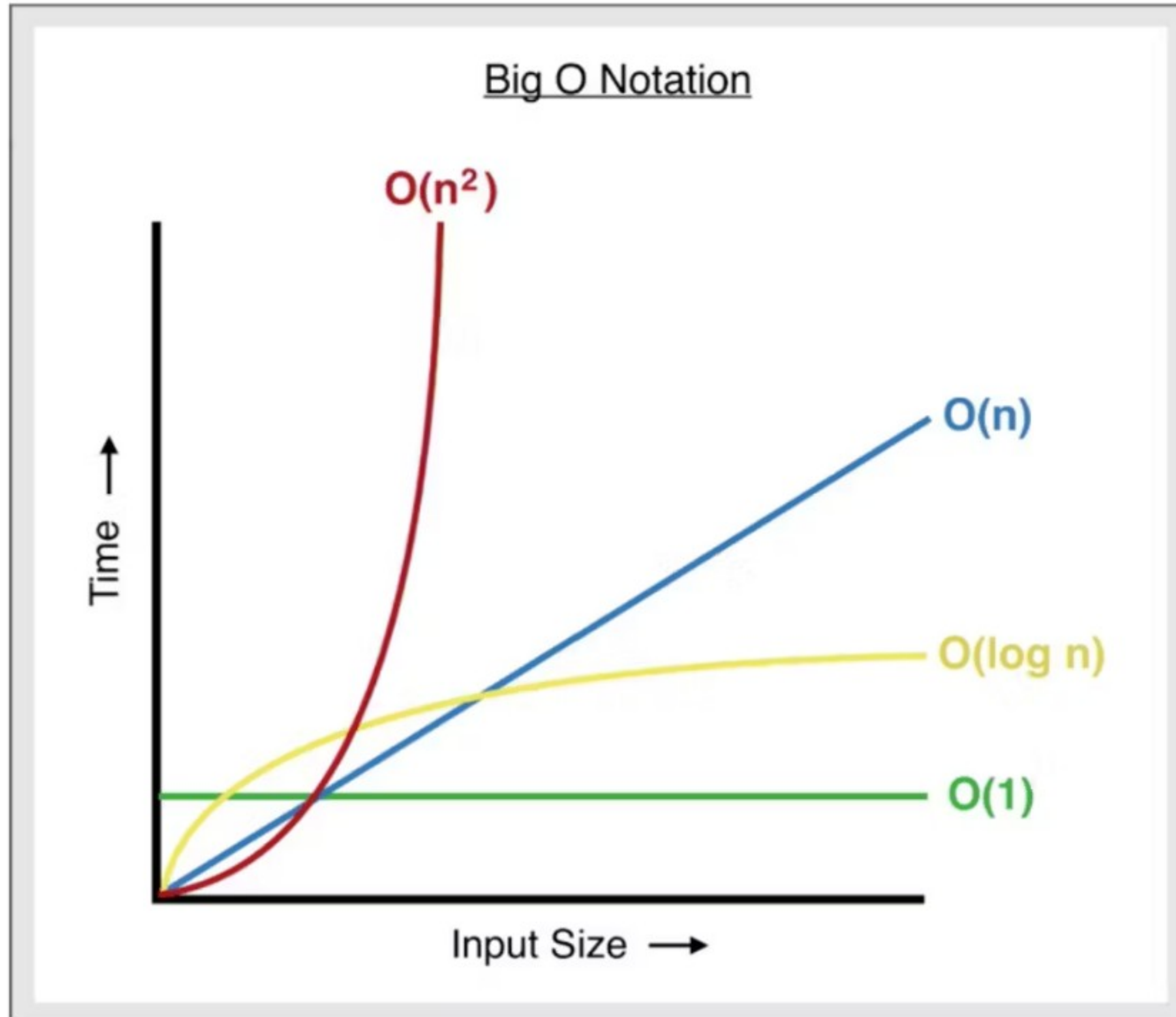
Binary Search

Search the number 7 in the array



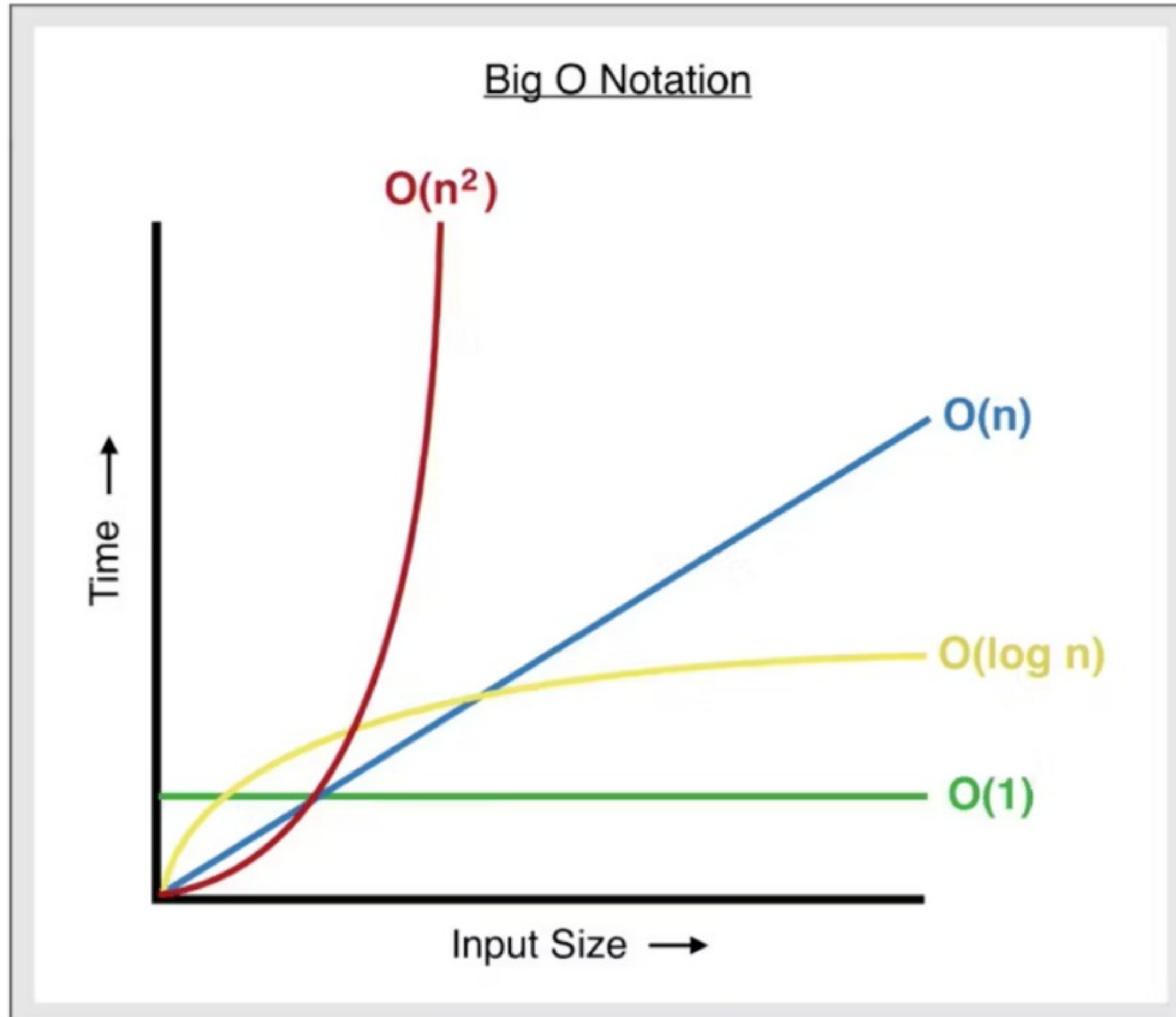
Visualiserring

<https://www.cs.usfca.edu/~galles/visualization/Search.html>



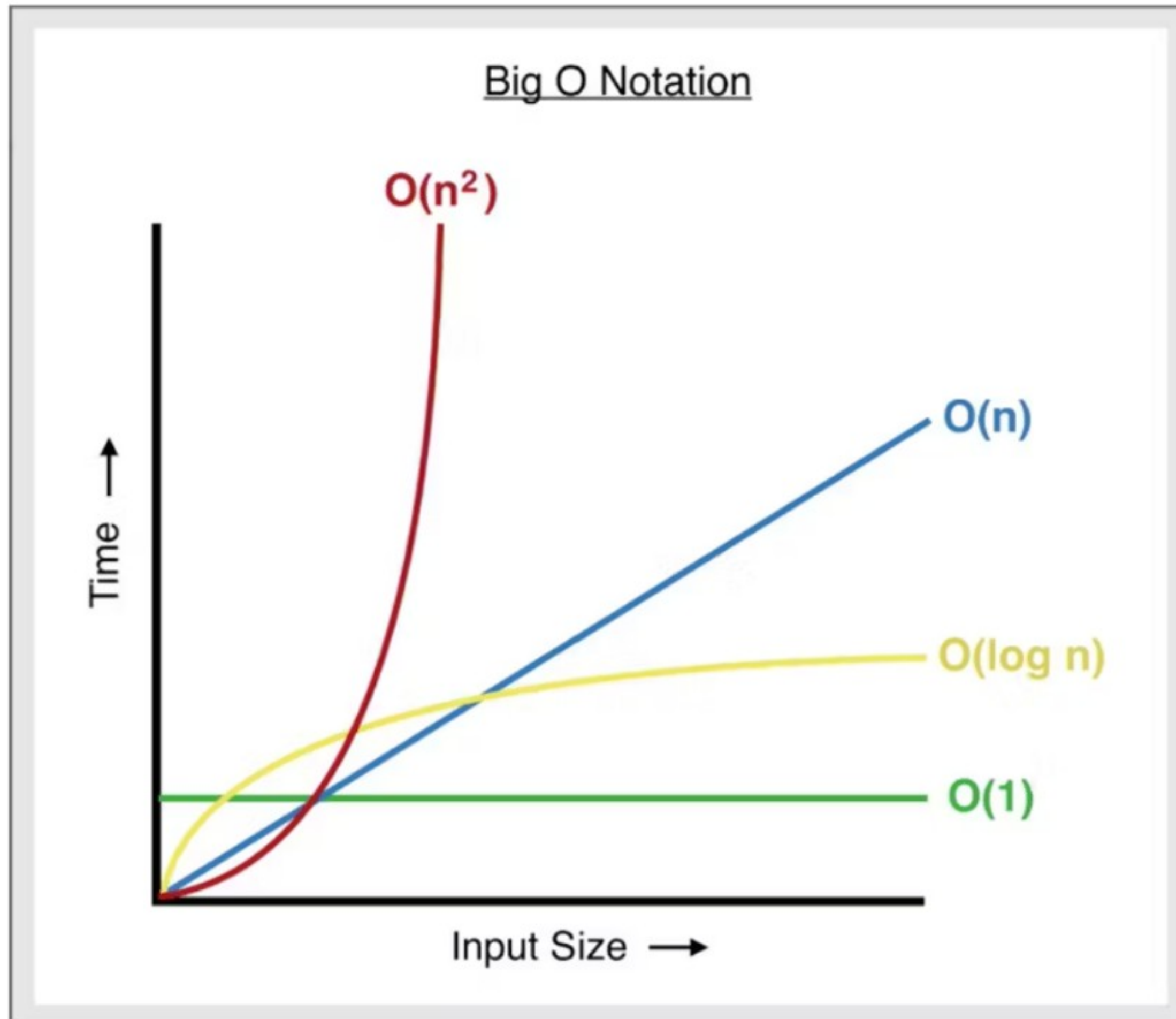
O-Notasjon

- Hvor vanskelig er et problem/algoritme å løse?
- AKA: Hvor mange steg trenger man for å løse problemet?
- Eksempel: Hvor mange steg må man gjøre for å finne en vilkårlig person i klassen?(Rett-Frem søk)



Samme eksempel, men med binærsøk

- Hvor stort er søkeromet?
- Hva er tidskompleksiteten?



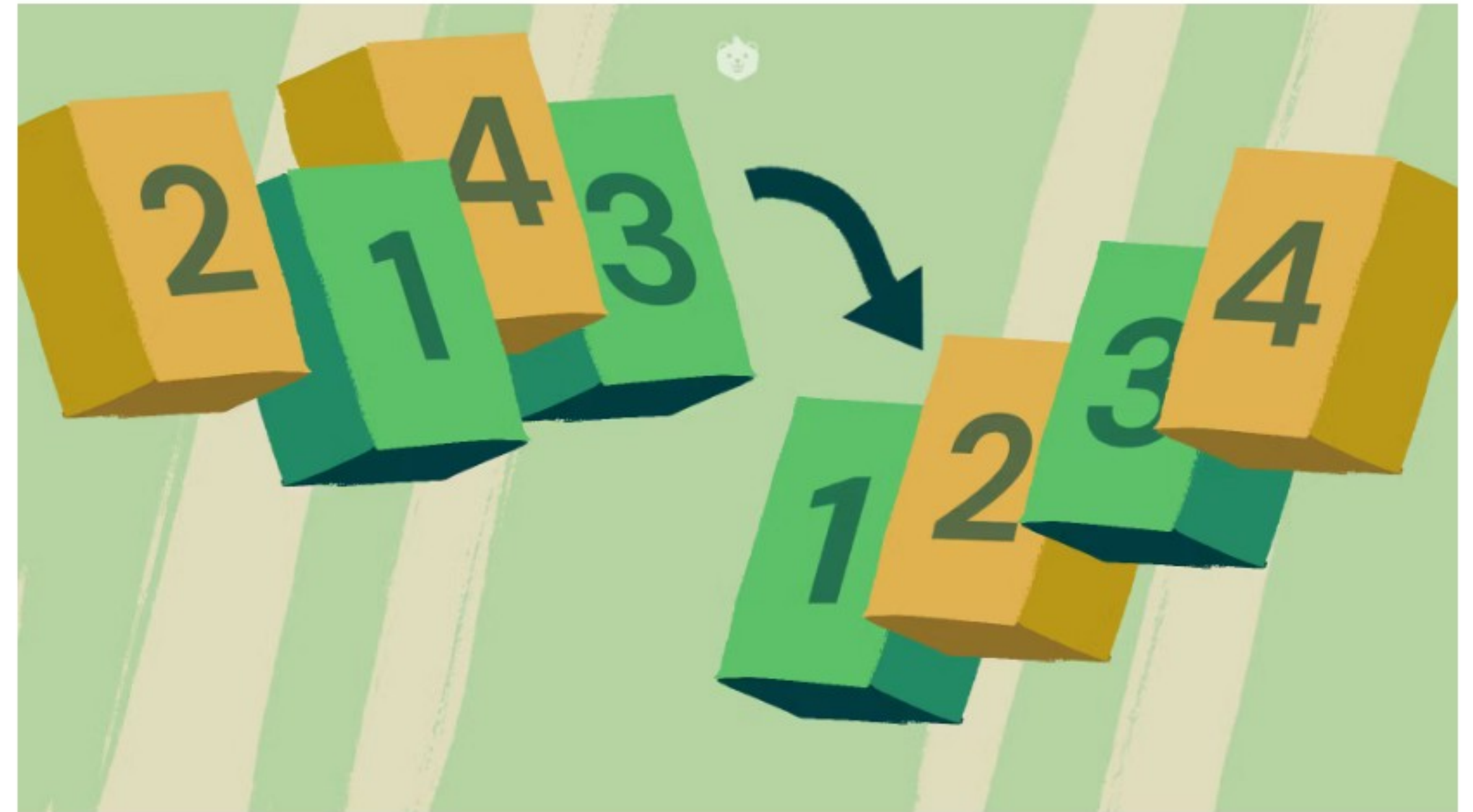
Bestemme O

- $O(n^3 + 50n^2 + 10000)$
- $O((n + 30) * (n + 5))$
- $O(n \log(n) + \log(n) \log(n))$
- $O(n + n + n + n + n + n)$

Bestemme O

- $O(n^3 + 50n^2 + 10000) = O(n^3)$
- $O((n + 30) * (n + 5)) = O(n^2)$
- $O(n \log(n) + \log(n) \log(n)) = O(n * \log(n))$
- $O(n + n + n + n + n + n) = O(6n) = O(n)$

Sorting



Hva er sortering?

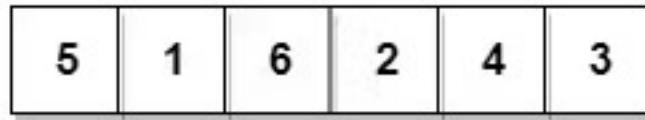
- Ordne elementer i en datastruktur
- Sette opp elementene i en ordnet rekkefølge
- NB: Elementene må være bevart

Hvorfor sortere?

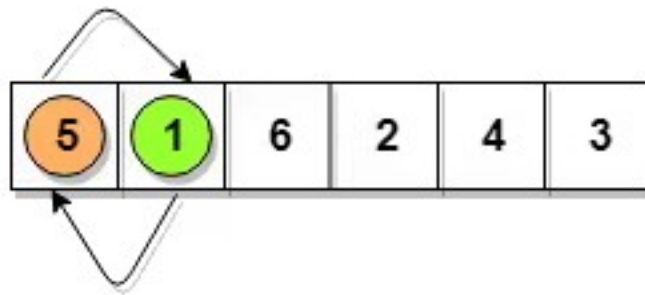
Flere problemer krever sorterte inputs(Binær søk)

Kommer til å få bruk av sortering i mange tilfeller

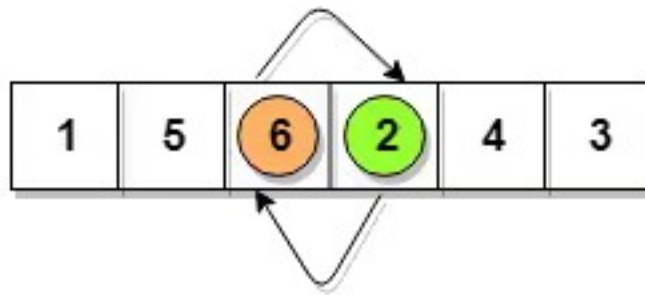
$5 > 1$
so interchange



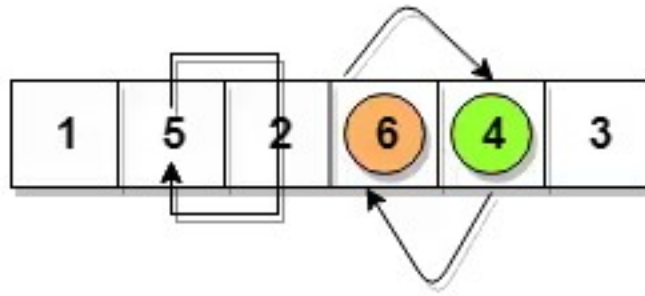
$5 < 6$
No swapping



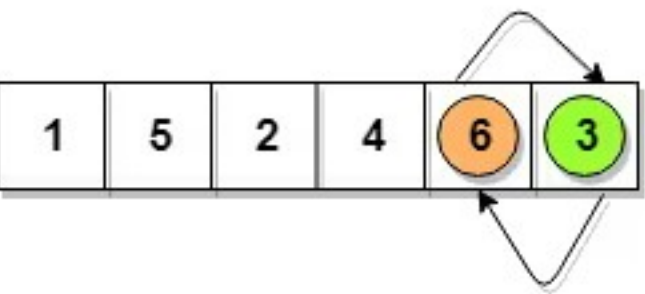
$6 > 2$
so interchange



$6 > 4$
so interchange



$6 > 3$
so interchange



This is first insertion

similarly, after all the
iterations, the array
gets sorted

Bubble sort

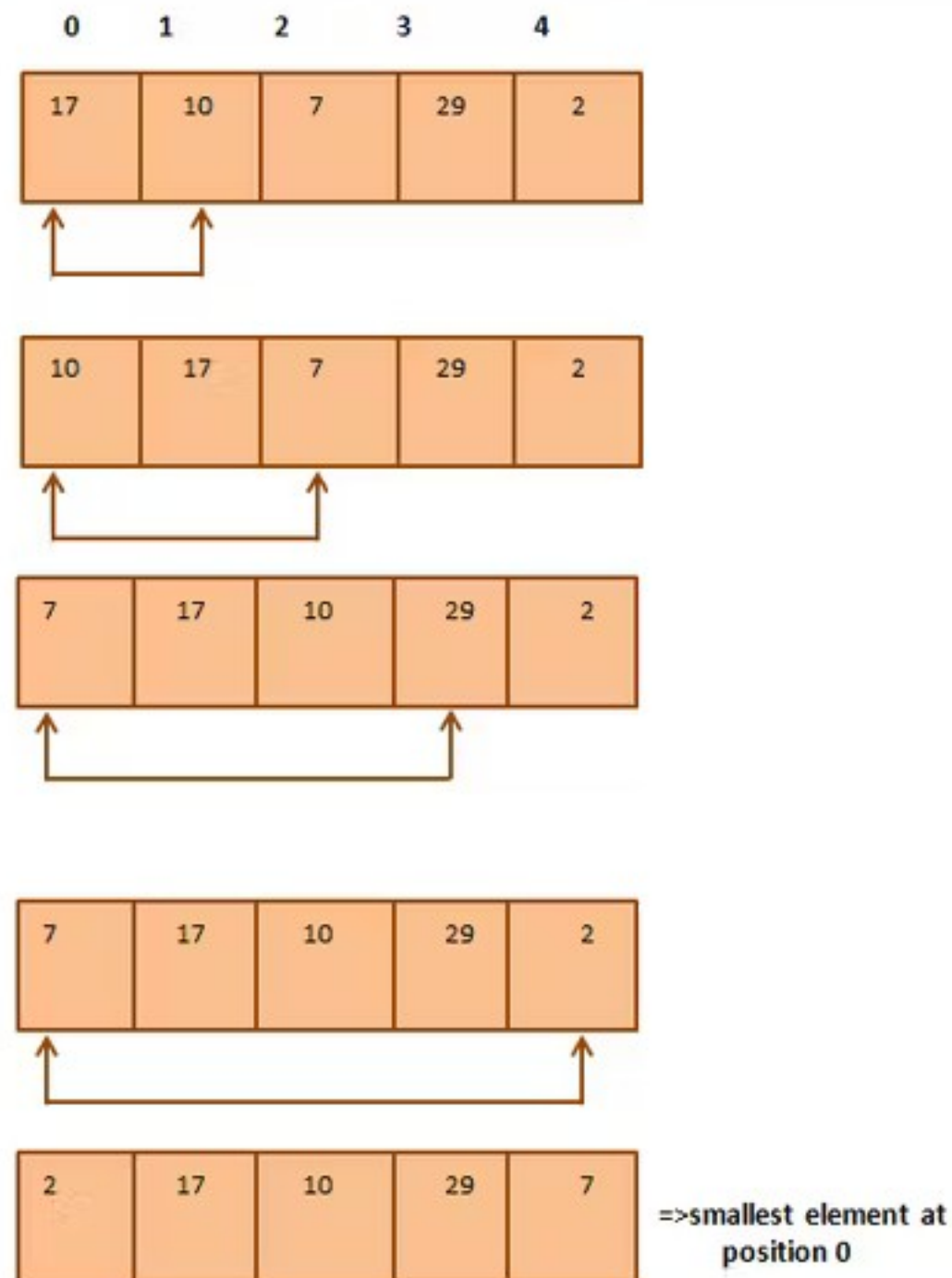
- Ide: Går gjennom lista og fikser feil
- Går parvis gjennom elementet
- Hvis det andre elementet er mindre enn det første, bytt plass

Bubble sort visualisering

<https://visualgo.net/en/sorting>



Pass 1:



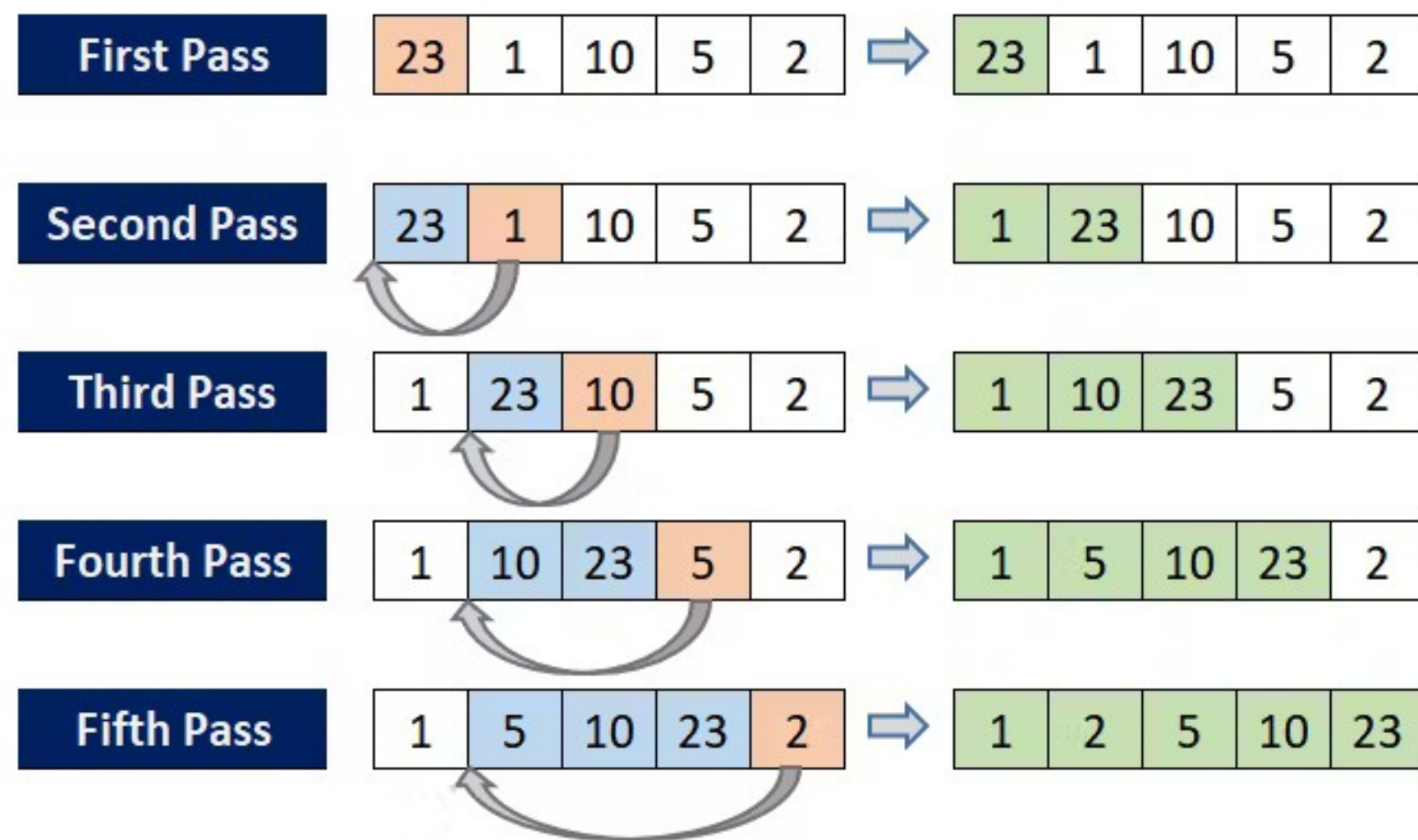
Selection sort

- Ide: Finne det minste elementet i lista, plasser det fremst
- For hver iterasjon: Bytt "fremst" peker

Selection sort visualisering

<https://visualgo.net/en/sorting>





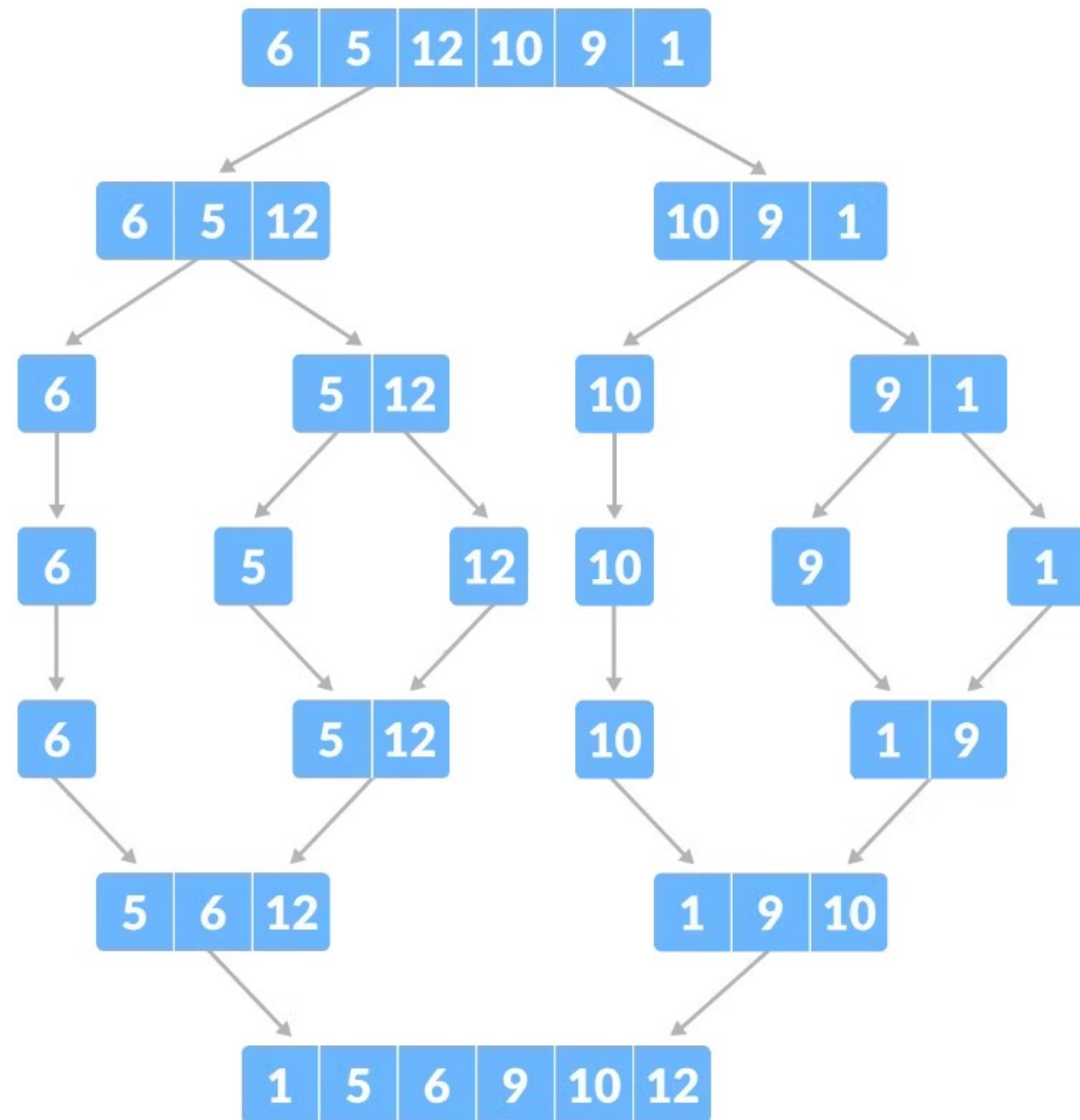
Insertion sort

- Ide: Legger elementene sortert i en liste
- Blir sortert når de "legges inn"
- Tar utgangspunkt i en posisjon og holder alt på venstresiden sortert

Insertion sort Visualisering

<https://visualgo.net/en/sorting>





Merge Sort

- Ide: Dele en array i 2 like store arrays
- Fortsetter rekursivt
- Når arrayet er inneholder 1 element, så kan du sammenligne verdiene og sortere
- Dette er det som menes med merging

Merge sort visualisering

<https://visualgo.net/en/sorting>

NB: Ikke like intuitivt

Gruppeoppgaver/Eksamen H2021(T/F)

For hver av påstandene nedenfor kan du anta at A er et array med n elementer, og at i er et heltall $0 \leq i < n$.

Ta utgangspunkt i Bubble, Selection og Insertion sort:

- (a) Etter x iterasjoner av den ytre loopen i ##### sort, er de x første elementene sortert.
- (b) Etter x iterasjoner av den ytre loopen i ##### sort, er de x siste elementene sortert.
- (c) ##### sort bytter kun elementer som står direkte ved siden av hverandre.
- (d) ##### sort garanterer et minimalt antall bytter.



Spørsmål

0 questions

0 upvotes