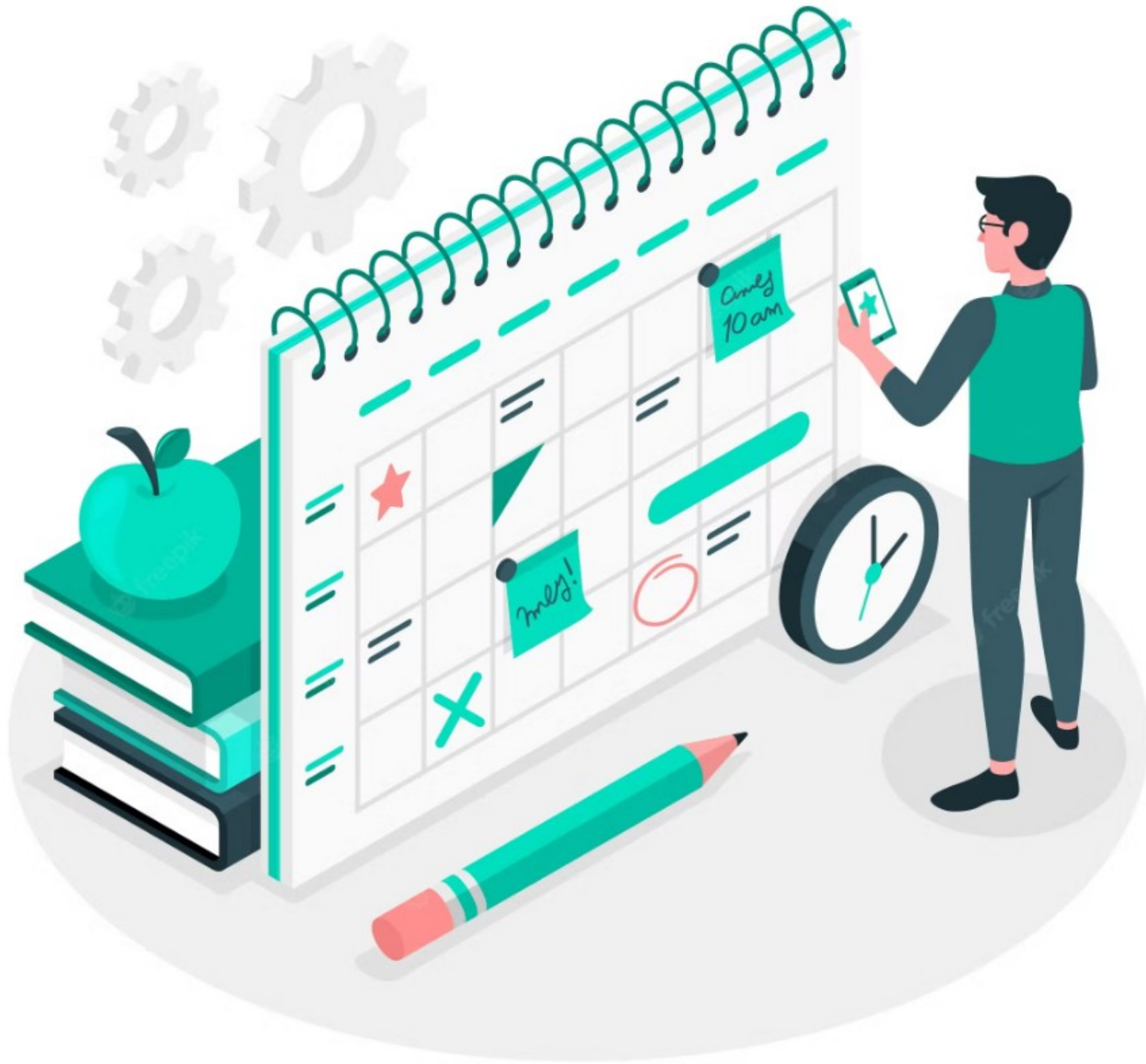


IN2010 - Gruppe 5

Uke 3 - Prioritetskøer - Binære Haps - Huffman-koding



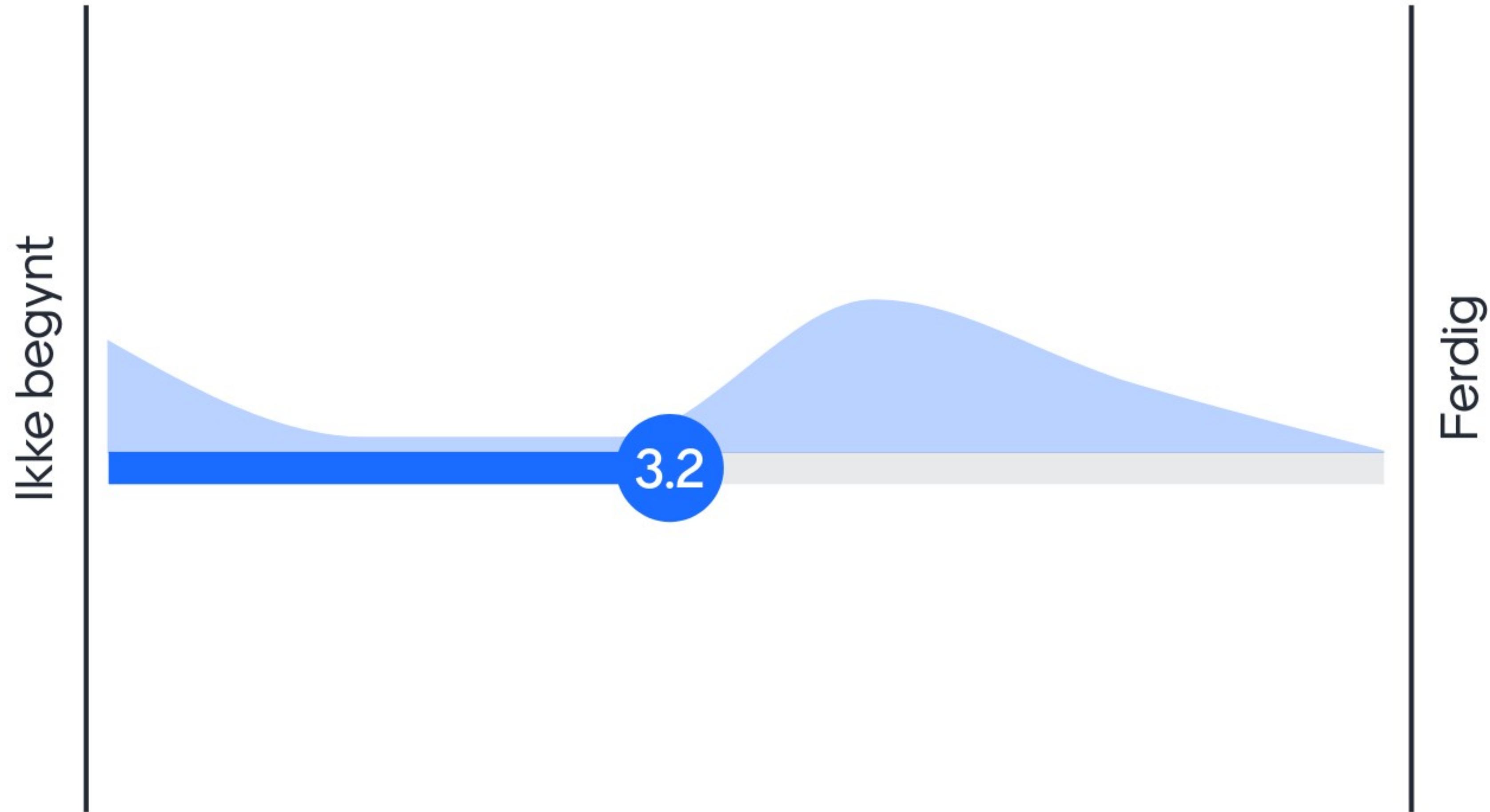
Bli med!



Dagens Plan

- Oblig 1 Update
- Pensumæ-gjennnomgang
- Gruppeoppgaver

Hvordan ligger derre ann med Oblig 1?



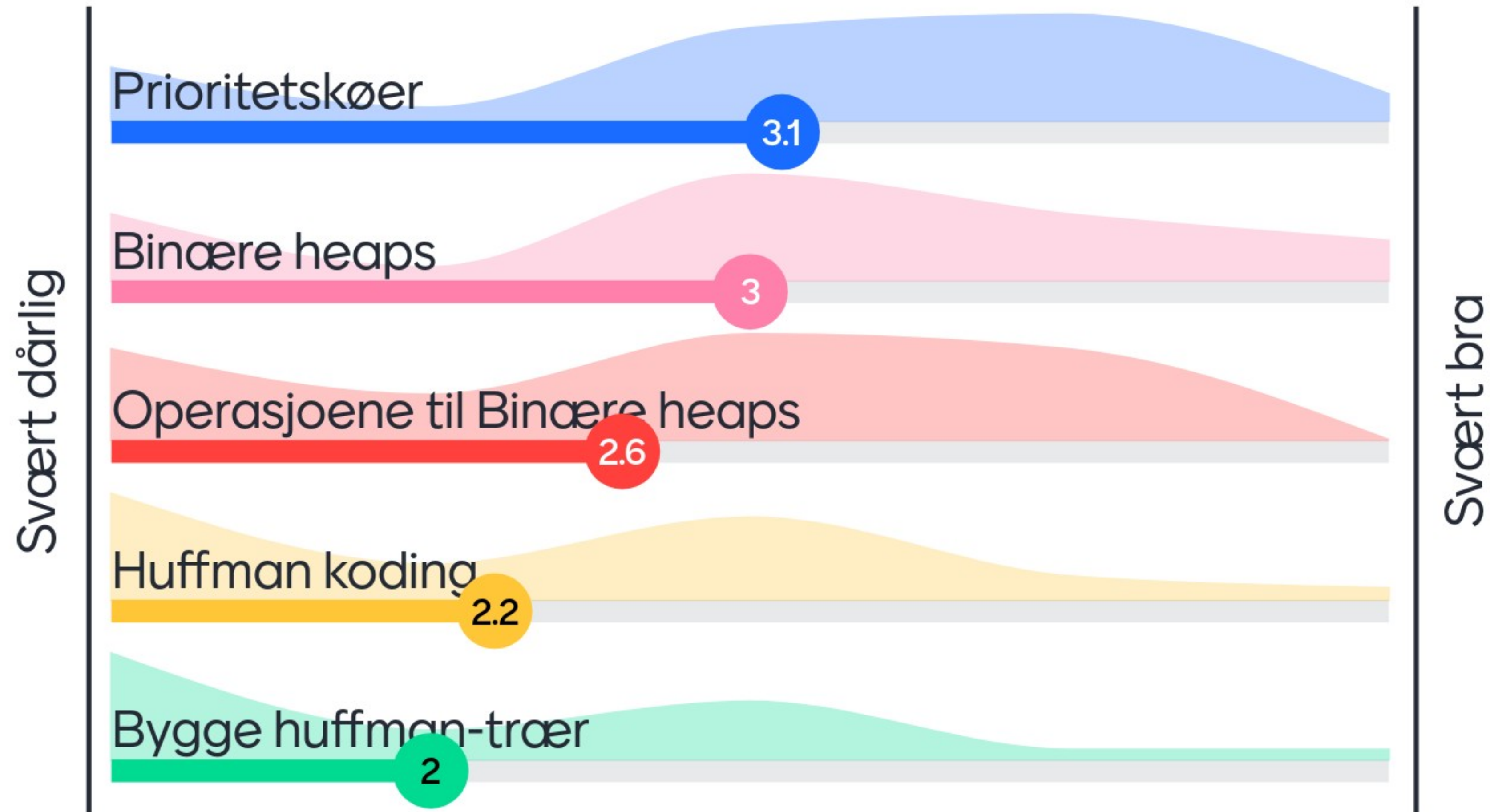
Spørsmål angående Obligen?

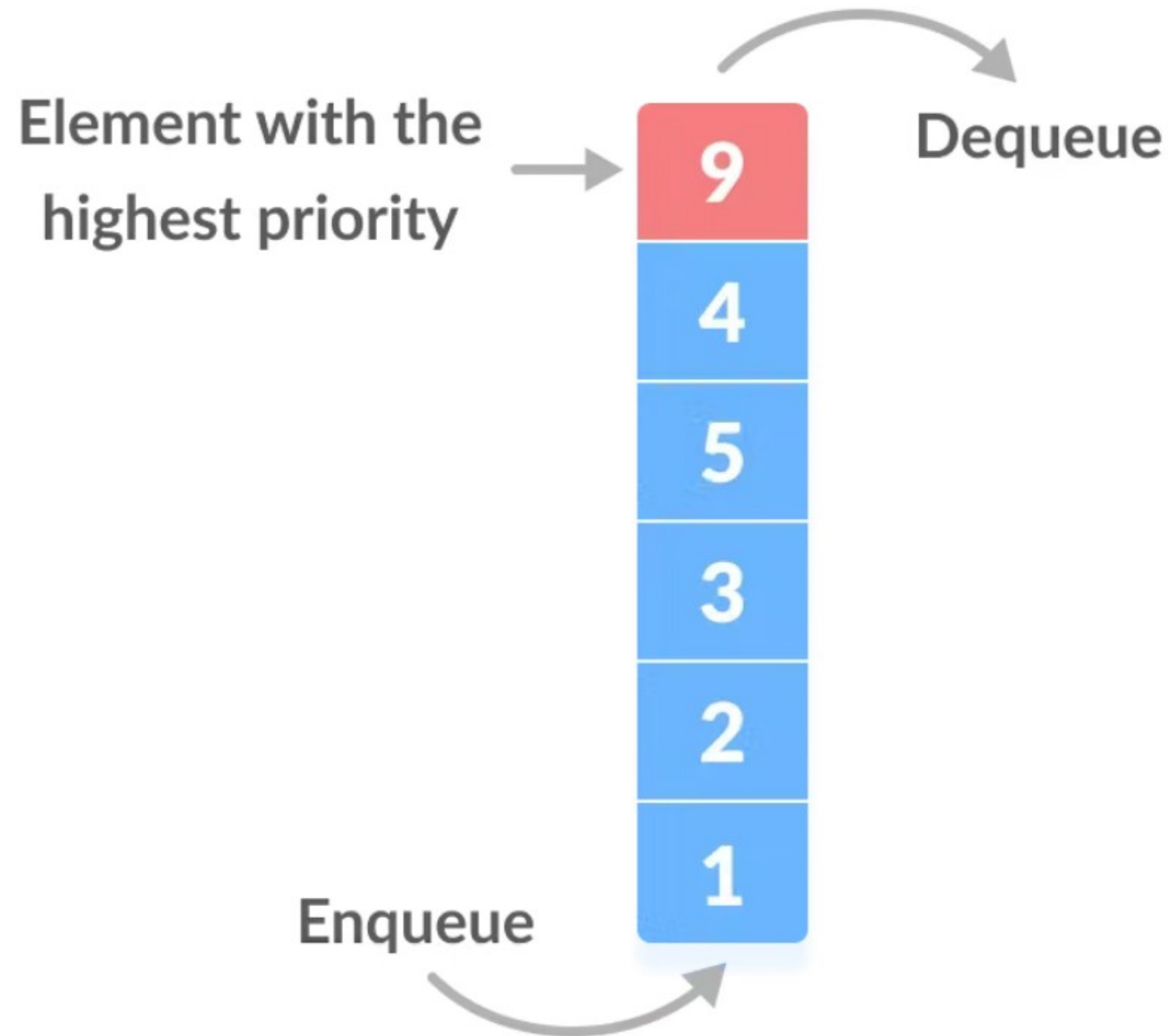
Noe som er uklart?

Pensum-
gjennomgang



Hvor godt forsto du ukens pensum?

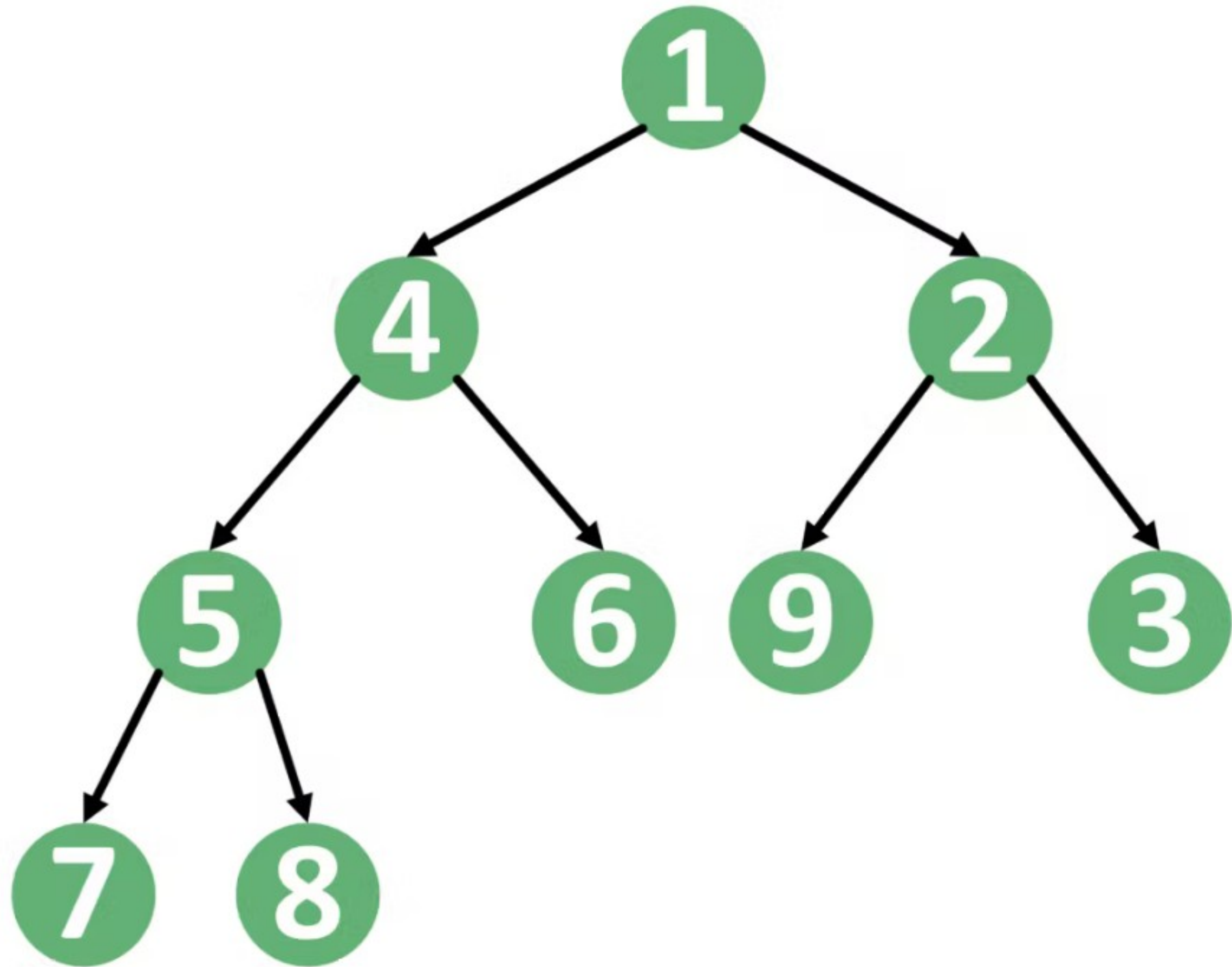




Prioritetskøer

- En kø/samling med elementer som er sortert etter prioritet
- Størrelse, alder, høyde, osv.
- En prioritetskø må støtte følgende operasjoner:
- `Insert(e)/pop(e)` - legger til et nytt element `e`
- `removeMin()/pop()` - Fjerner elemente med høyest prioritet

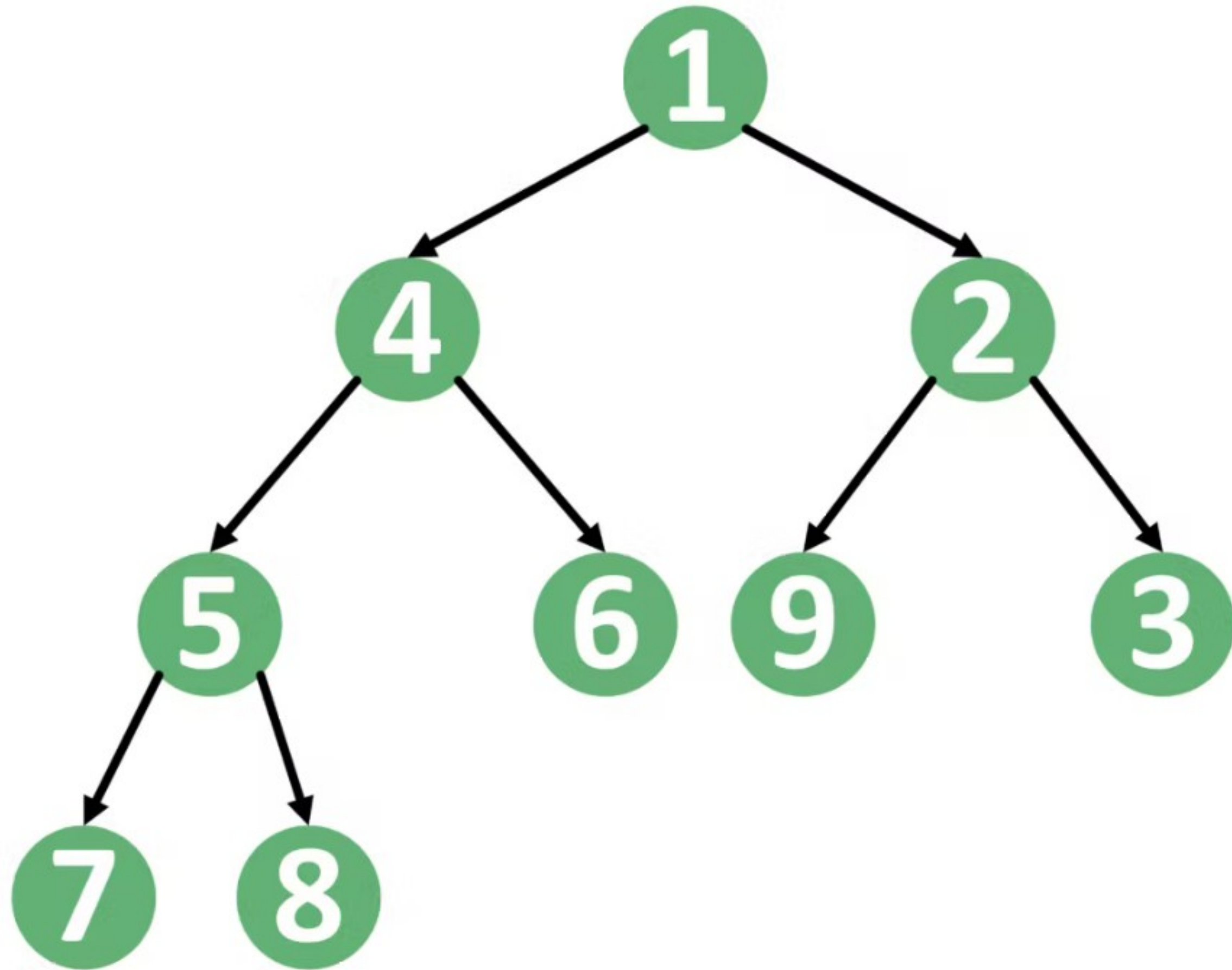




Eksempel

Binære heaps
(min heaps)

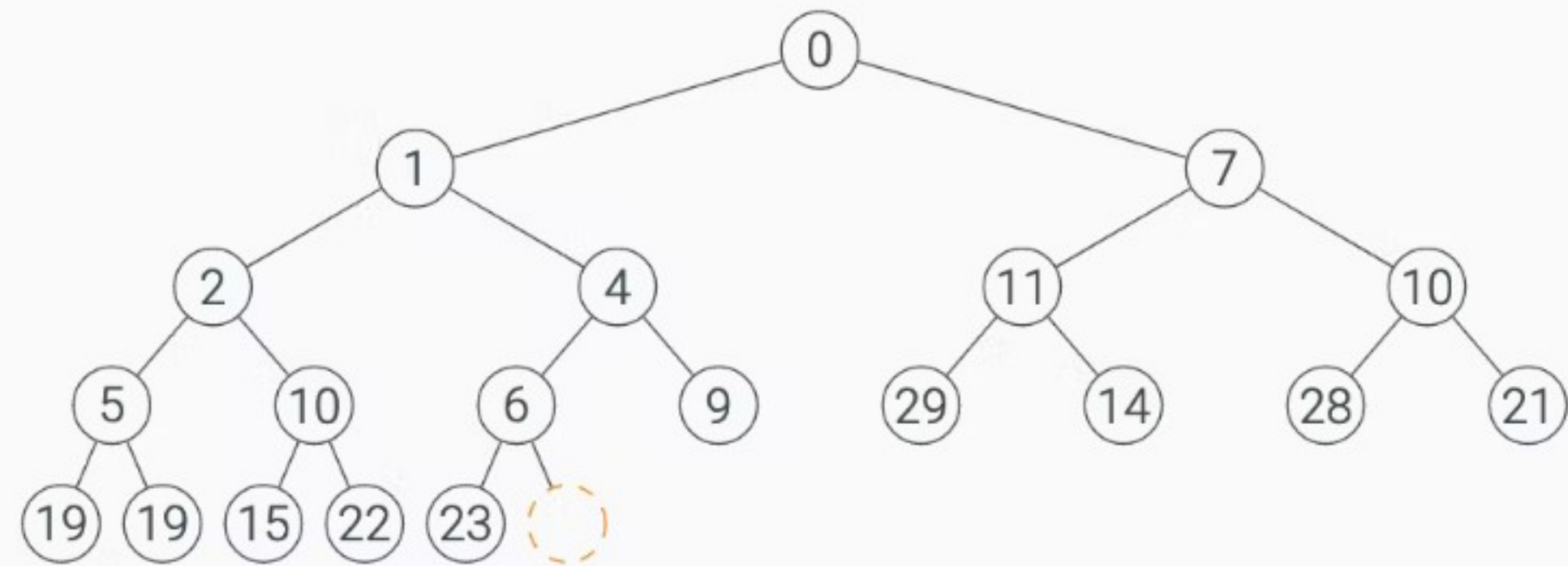




Binære Heaps

1. Hver node v (ikke rotnoden) er større enn foreldre node
2. Treet må være komplett
3. Det betyr at treet fylles opp fra høyre til venstre





0	1	7	2	4	11	10	5	10	6	9	29	14	28	21	19	19	15	22	23	-
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Binære heaps: Operasjoner

Ideen: Tar alltid utgangspunkt i neste "ledige" plass(innsetting)

Eller så tar vi utgangspunkt i det siste elementet(sletting)

Binære Heap- Demonstrasjon

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



ALGORITHM: INNSETTING I HEAP

Input: Et array A som representerer en heap med n elementer, og et element x

Output: Et array som representerer en heap, som inneholder x

```
1 Procedure Insert( $A, x$ )
2    $A[n] \leftarrow x$ 
3    $i \leftarrow n$ 
4   while  $0 < i$  and  $A[i] < A[\text{ParentOf}(i)]$  do
5      $A[i], A[\text{ParentOf}(i)] \leftarrow A[\text{ParentOf}(i)], A[i]$ 
6      $i \leftarrow \text{ParentOf}(i)$ 
```

ALGORITHM: FJERNING AV MINSTE ELEMENT FRA HEAP

Input: Et array A som representerer en heap med n elementer

Output: Et array som representerer en heap der minste verdi er fjernet

```
1 Procedure RemoveMin(A)
2    $x \leftarrow A[0]$ 
3    $A[0] \leftarrow A[n-1]$ 
4    $i \leftarrow 0$ 
5   while RightOf(i) <  $n - 1$  do
6      $j \leftarrow$  if  $A[\text{LeftOf}(i)] \leq A[\text{RightOf}(i)]$  then LeftOf(i) else RightOf(i)
7     if  $A[j] \leq A[i]$  then
8        $A[i], A[j] \leftarrow A[j], A[i]$ 
9        $i \leftarrow j$ 
10    else
11      break
12  if LeftOf(i) <  $n - 1$  and  $A[\text{LeftOf}(i)] \leq A[i]$  then
13     $A[i], A[\text{LeftOf}(i)] \leftarrow A[\text{LeftOf}(i)], A[i]$ 
14  return x
```

Binære heaps - Sletting



Huffman- koding

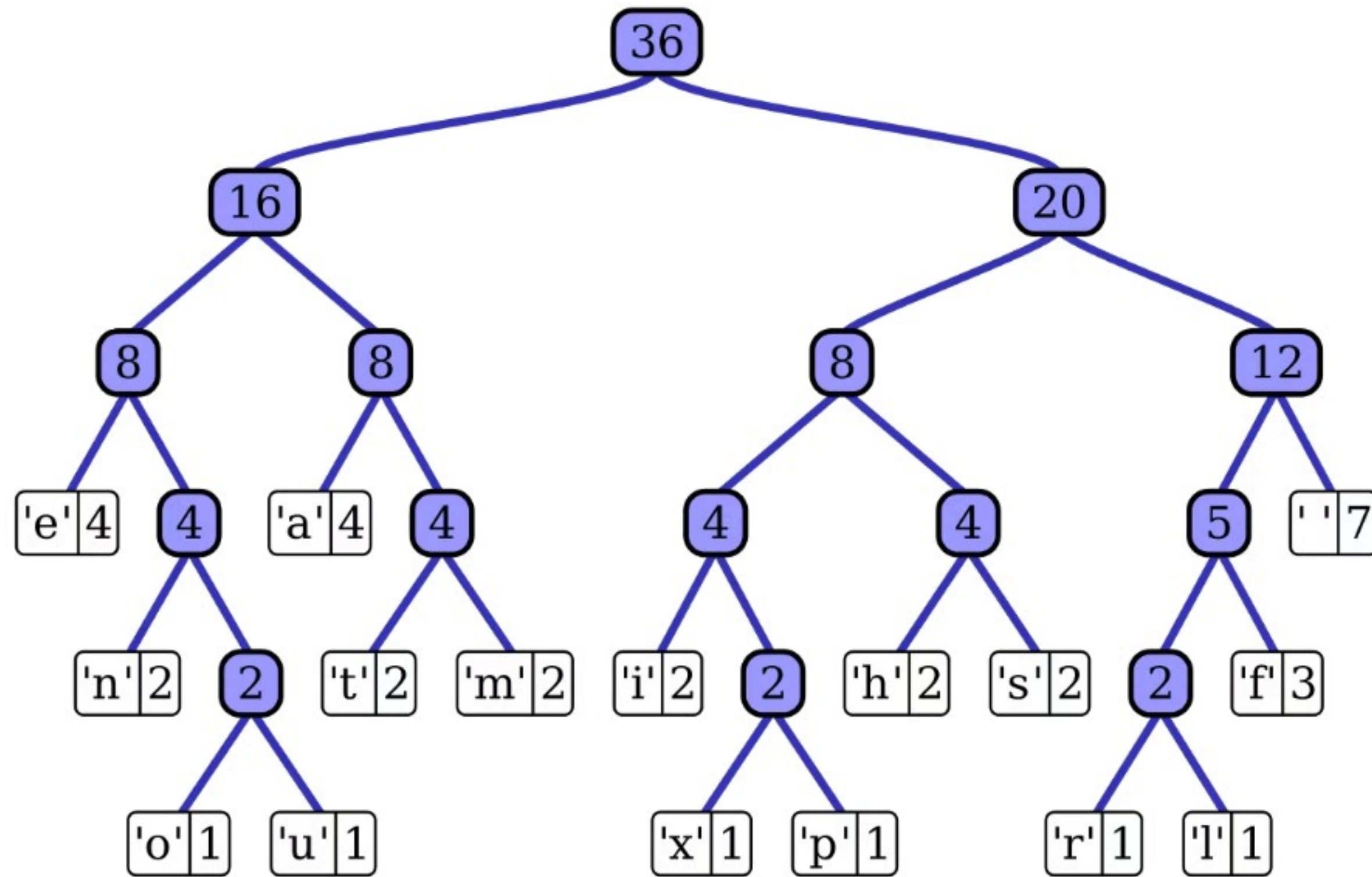


Characters	Code	Frequency	Total Bits
e	000	15	45
a	001	11	33
i	010	2	6
o	011	8	24
u	100	10	30
space	101	13	39
new line	110	5	15
Total		64	192

Huffman-koding

- Formål: Komprimere data
- Huffman-koding representerer frekvenser av symboler
- Med frekvensene så kan vi representere setninger med bitstrenger





Huffman træ

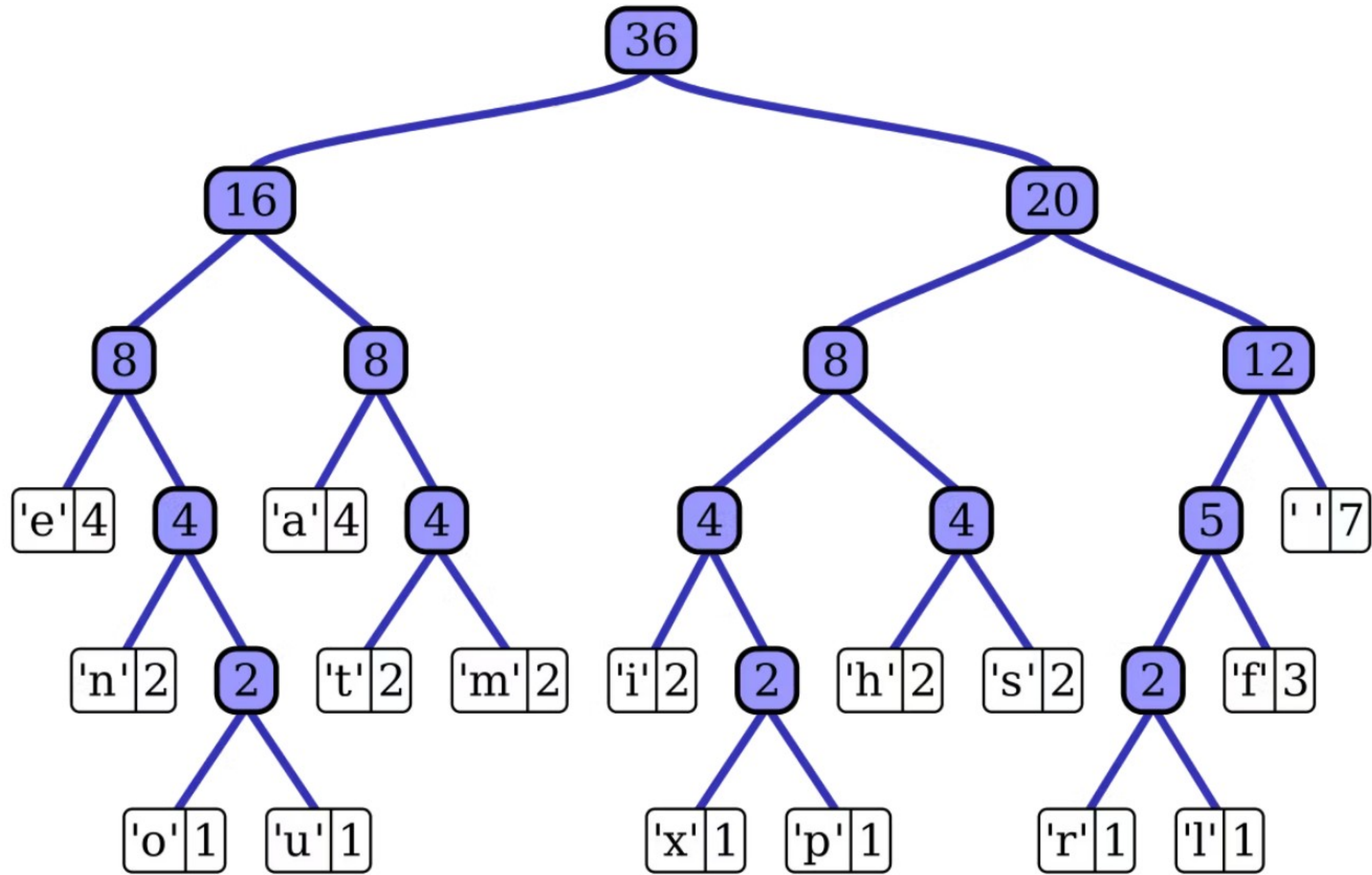
- Huffman træ er treet som viser frekvensene av symboler
- Start med å lage en frekvenstabell
- Lag en prioritetskø som prioriterer basert på frekvens
- Velg de to minste elementene fra køen
- Lag en ny node, som har de to nodene som bar
- Legg den nye noden tilbake i køen



Huffman træer - demo

<https://cmps-people.ok.ubc.ca/ylucet/DS/Huffman.html>





Re

ALGORITHM: BYGGE HUFFMAN TRÆR

Input: En mengde C med par (s, f) der s er et symbol og f er en frekvens

Output: Et Huffman-tre

```
1 Procedure Huffman( $C$ )
2    $Q \leftarrow$  new PriorityQueue
3   for  $(s, f) \in C$  do
4     Insert( $Q$ , new Node( $s, f, \text{null}, \text{null}$ ))
5   while Size( $Q$ ) > 1 do
6      $v_1 \leftarrow$  RemoveMin( $Q$ )
7      $v_2 \leftarrow$  RemoveMin( $Q$ )
8      $f \leftarrow v_1.\text{freq} + v_2.\text{freq}$ 
9     Insert( $Q$ , new Node( $\text{null}, f, v_1, v_2$ ))
10  return RemoveMin( $Q$ )
```



Gruppeoppgaver



Oppgave 14

A file contains only spaces and digits in the following frequency: space (9), a (5), b (1), d (3), e (7), f (3), h (1), i (1), k (1), n (4), o (1), r (5), s (1), t (2), u (1), v (1).

Construct the Huffman tree.

Ekstra: Hva er huffman-koden til strengen "hei du der borte"



Oppgave 15

A file contains only colons, spaces, newlines, commas, and digits in the following frequency: colon (100), space (605), newline (100), comma(705), 0 (431), 1 (242), 2 (176), 3 (59), 4 (185), 5 (250), 6 (174), 7 (199), 8 (205), 9 (217).

Construct the Huffman tree

Ekstra: Hva er huffman-koden til telefonnummeret ditt?



Ukesoppgaver

- Implementer algoritmene fra forelesningen (uke 36).
- Bullet 2
- Bullet 3



Ask me anything

2 questions
1 upvotes