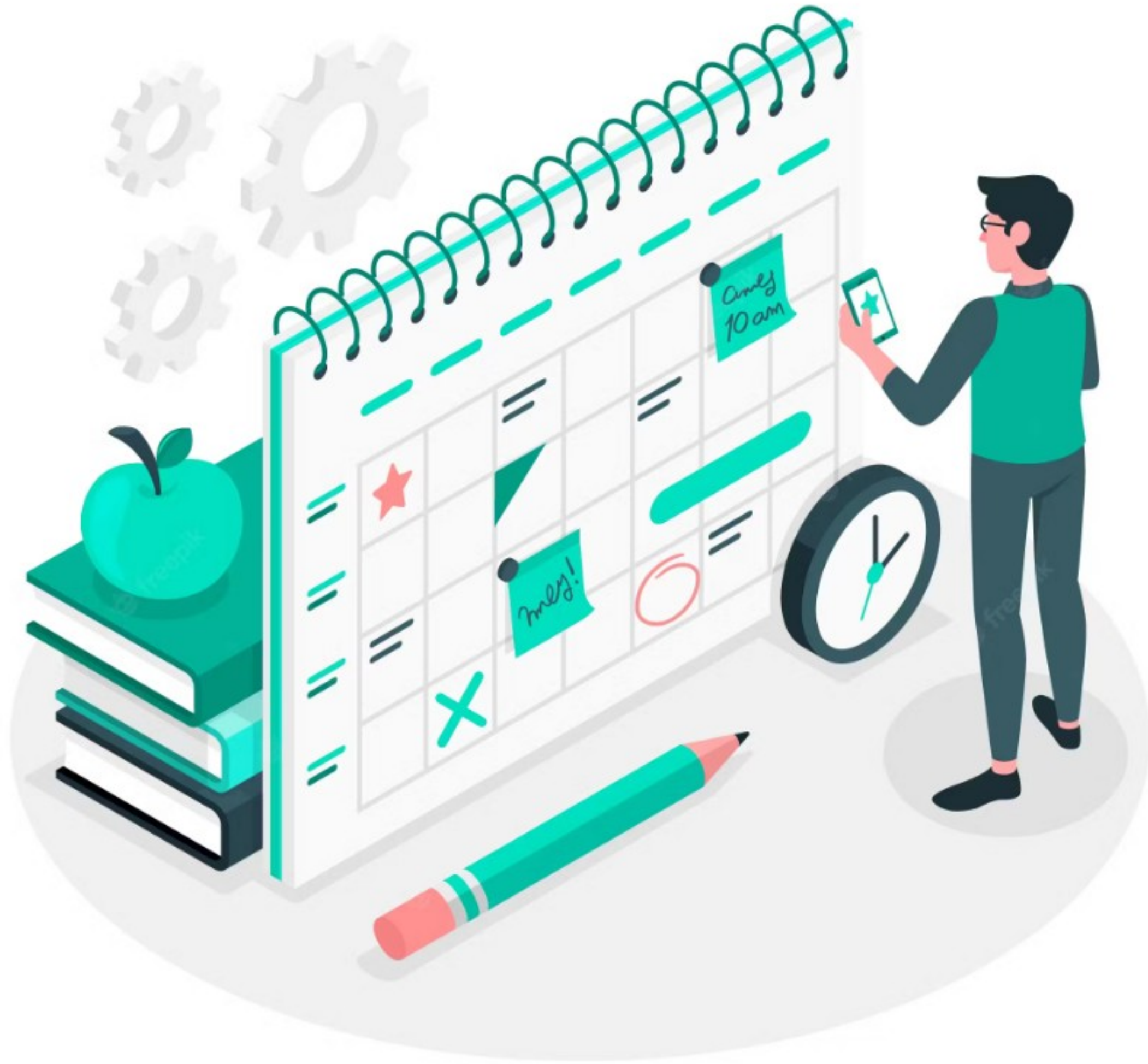


IN2010 Gruppe 4

Uke 13 - Repetisjon: Prioritetskøer



Bli med



Dagens Plan

- Eksamentips
- Gjennomgang(Heaps)
- Felles øvelser
- Pause
- Gjennomgang(Huffmankoding)
- Gruppeoppgaver

Eksamentips

- Faget krever forståelse
- Ikke overkompliser "Kode" aspektet
- Det er mange syntaktiske aspekter som må være på plass i en reell implementasjon, Disse må man ikke ta like stor hensyn til
- Pseudokode handler om å beskrive steg
- Om du klarer å løse problemet manuelt -> Beskriv stegene du gjorde for å løse det
- Se etter ting som kjennetegner datastrukturer og algoritmene vi har lært
- Oppgavebeskrivelsen vil ofte nevne hints som illustrerer et tilfelle som vil passe best for en gitt algoritme
- Ren implementasjon av en algoritmen er sjeldent svaret
- Løsningene krever ofte bruk av en algoritme for å løse problemet -> Men selve løsningen vil også kreve noen ekstra steg
- Fokuser på hva PROBLEMET er og hvordan man LØSER det, før man tenker på hvilke algoritmer som passer best
- Ikke bruk for mye tid til å finne en "bedre løsning"
- Eksamen er 4 timer, og man kan alltid finne en "bedre løsning". Kom heller tilbake til oppgaven senere

Gjennomgang

Hva er vanskelig med heaps?

alt

Sletting

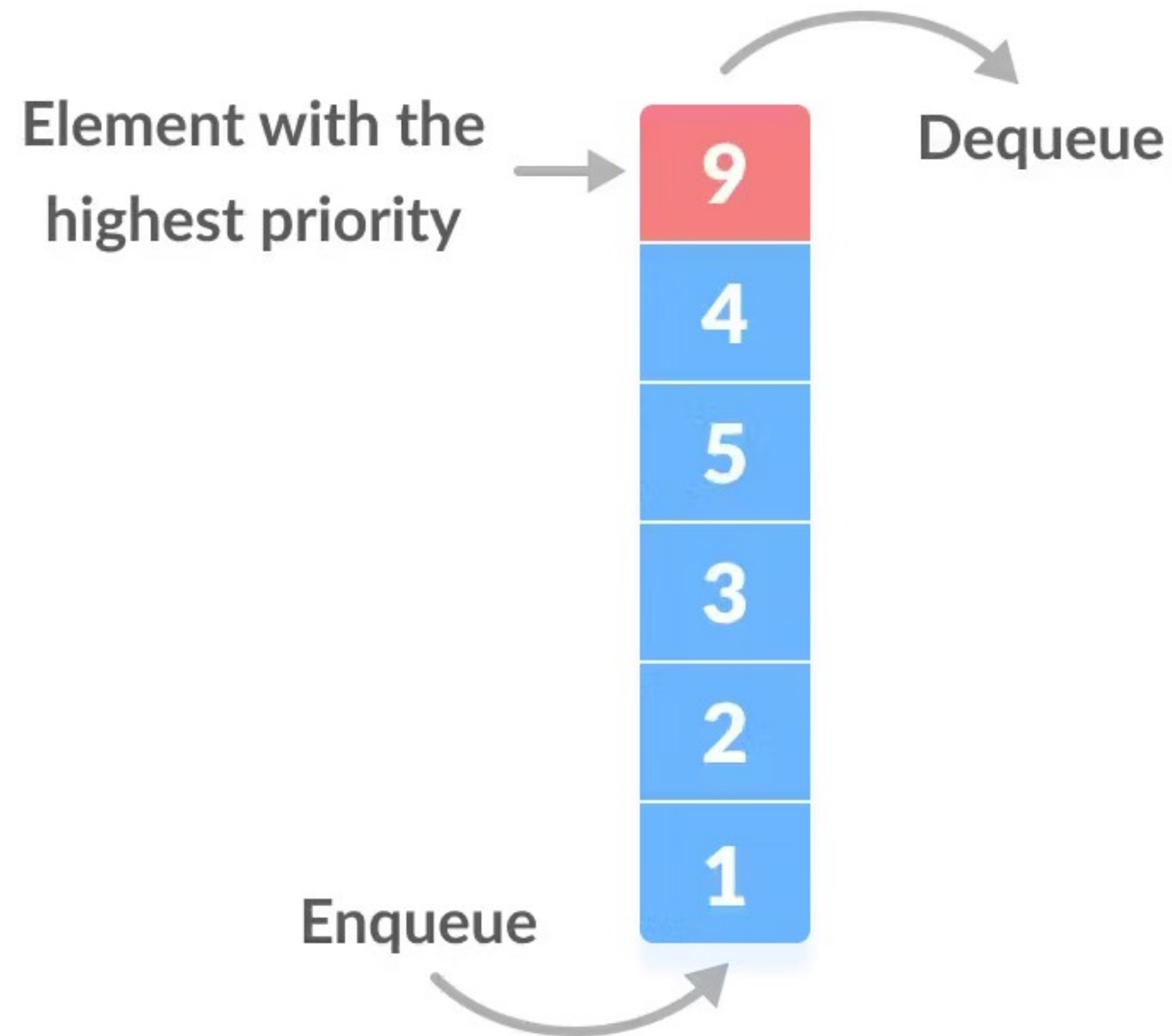
sletting i linerprobing

Fjerne element

forskjellen mellom det og
trær

bubble down

resice

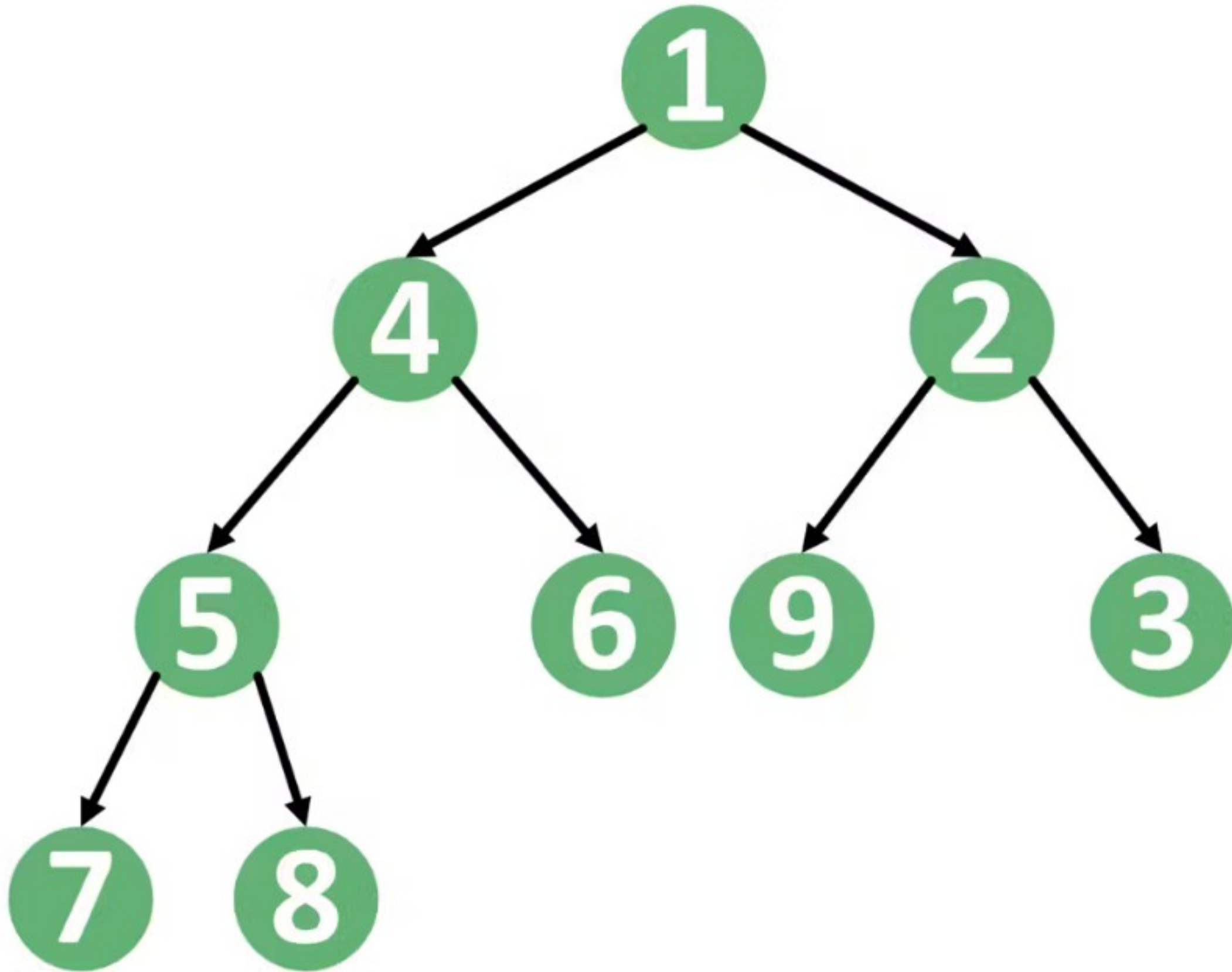


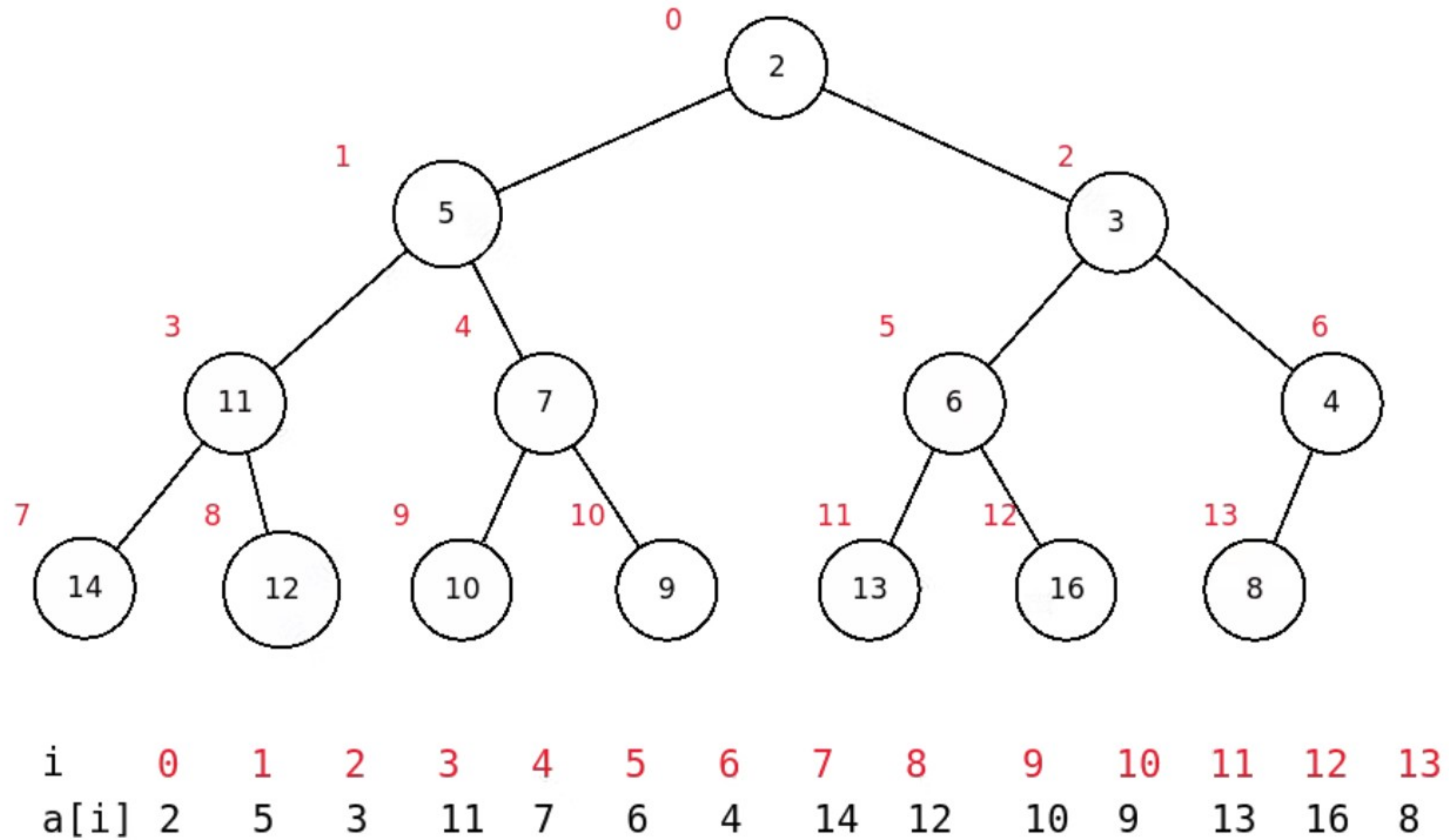
Prioritetskøer

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

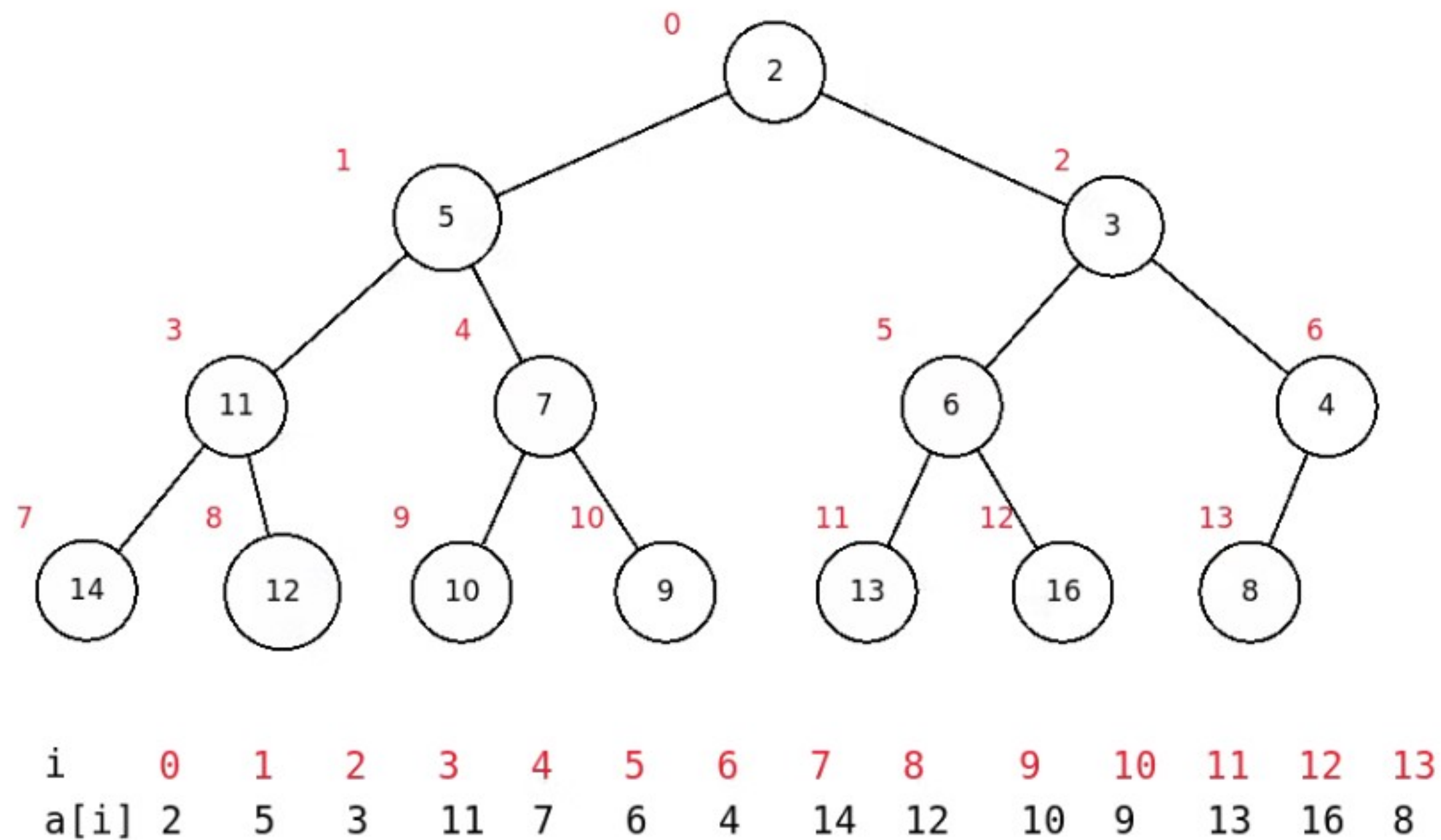
Binære Heaps

1. Hver node v er mindre enn barne-nodene
2. Treet må være komplett
3. Det betyr at treet fylles opp fra venstre til høyre





Array Representasjon



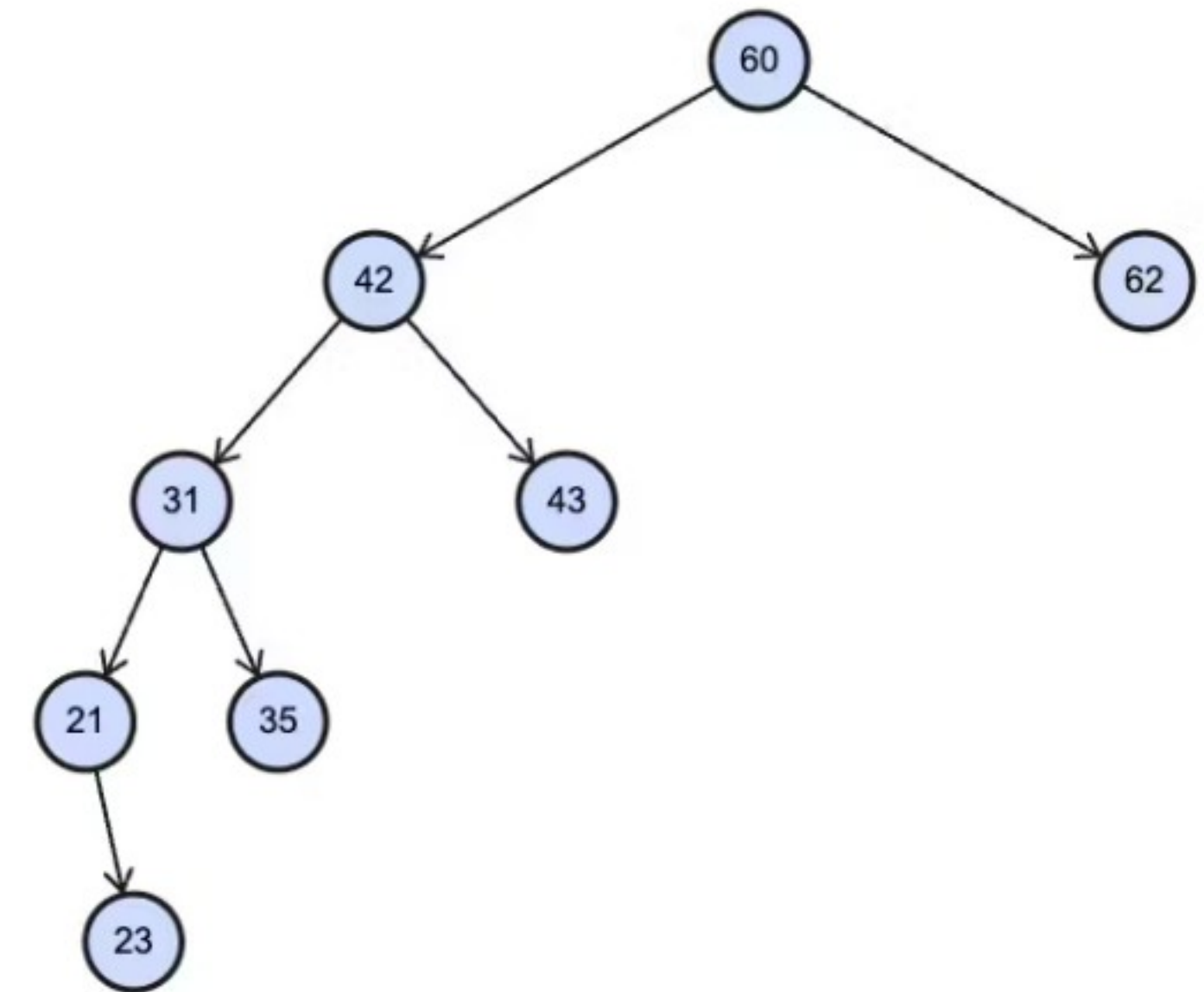
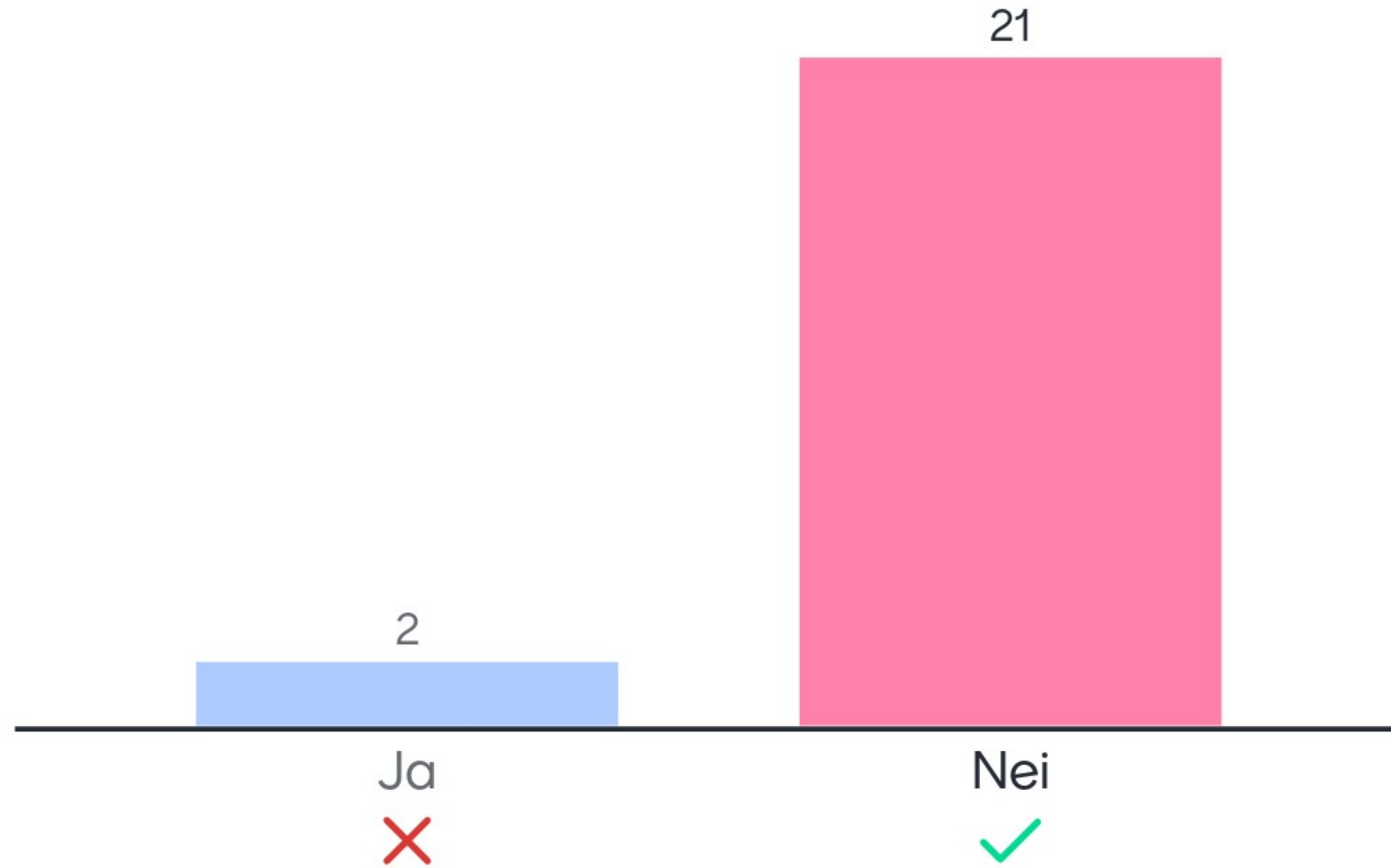
Array Representasjon

- Siden heaps fylles fra høyre til venstre så vet vi følgende:
- Forelderen til en gitt node på index i vil alltid ligge på index $(i-1)//2$
- Barnene til en node på index i vil alltid ligge på index $(2i + 1)$ og $(2i + 2)$

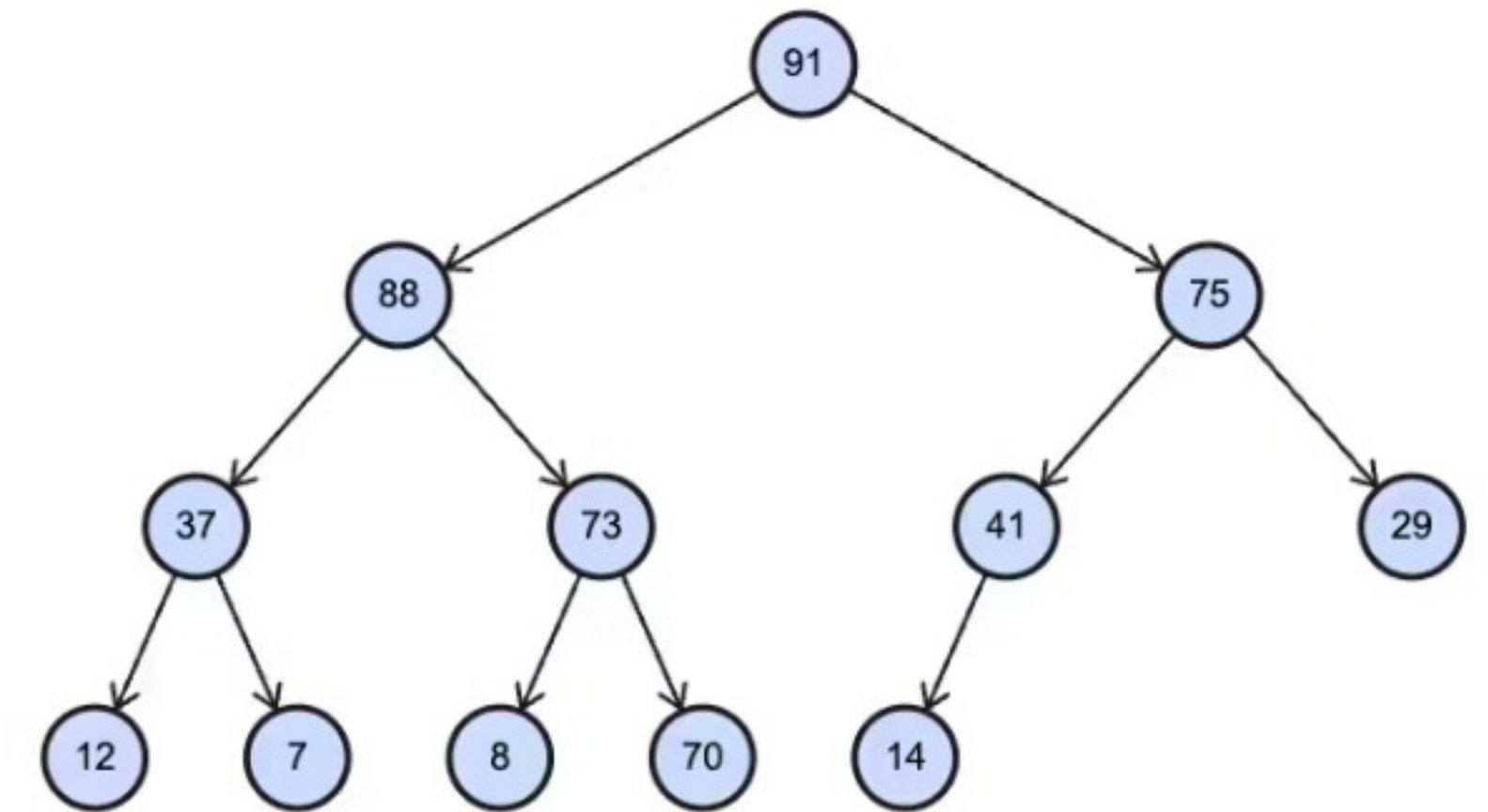
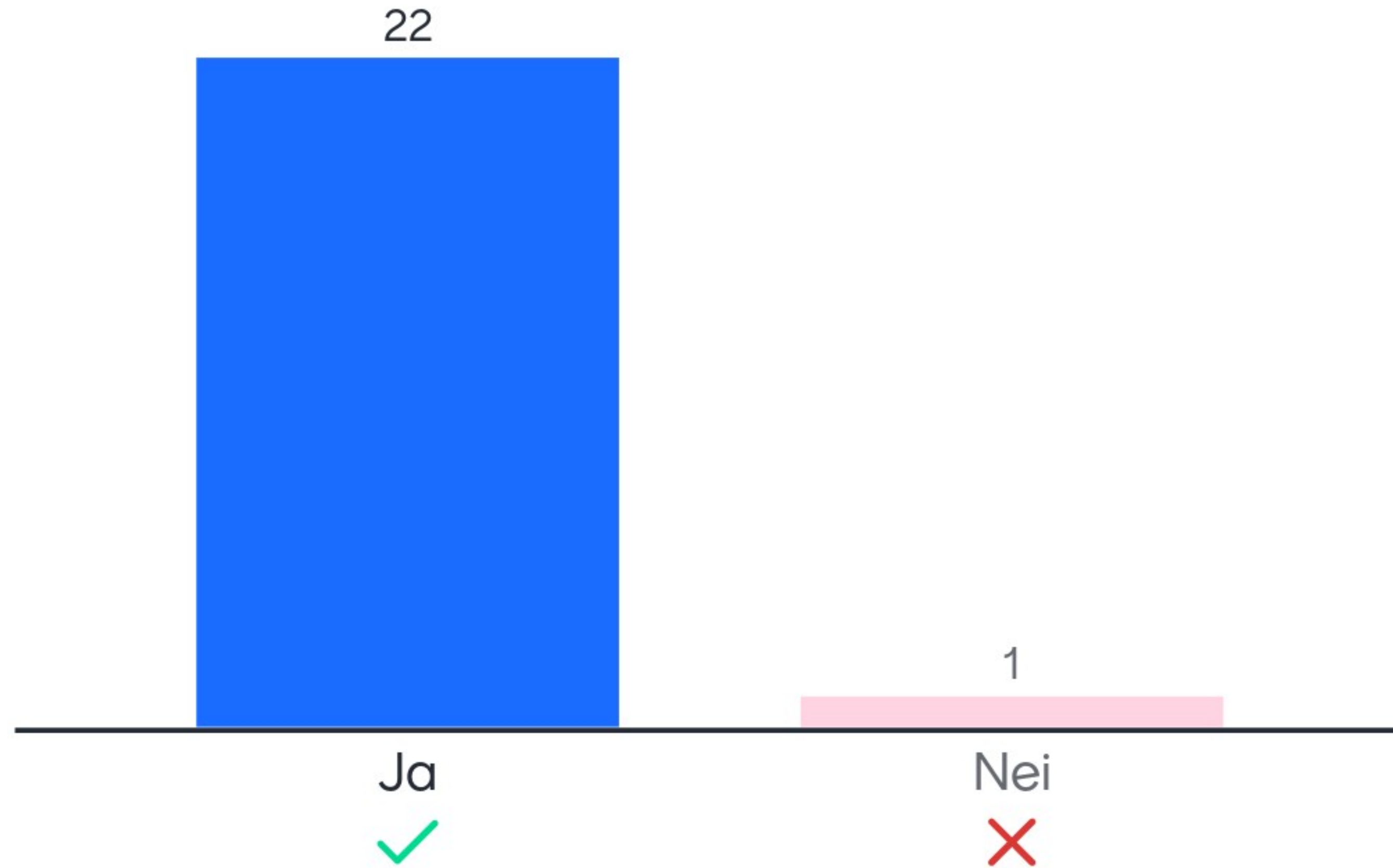
Quiz



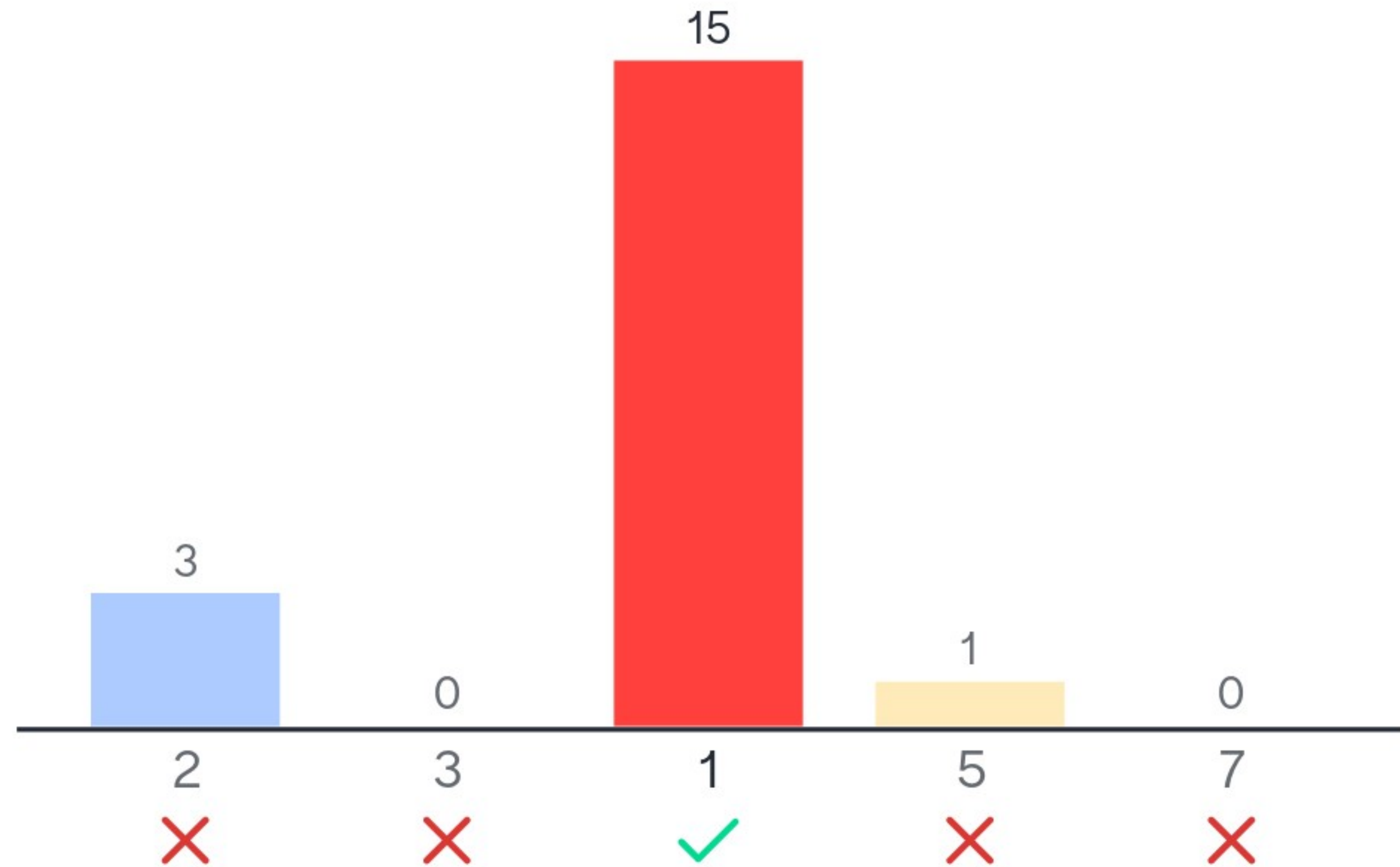
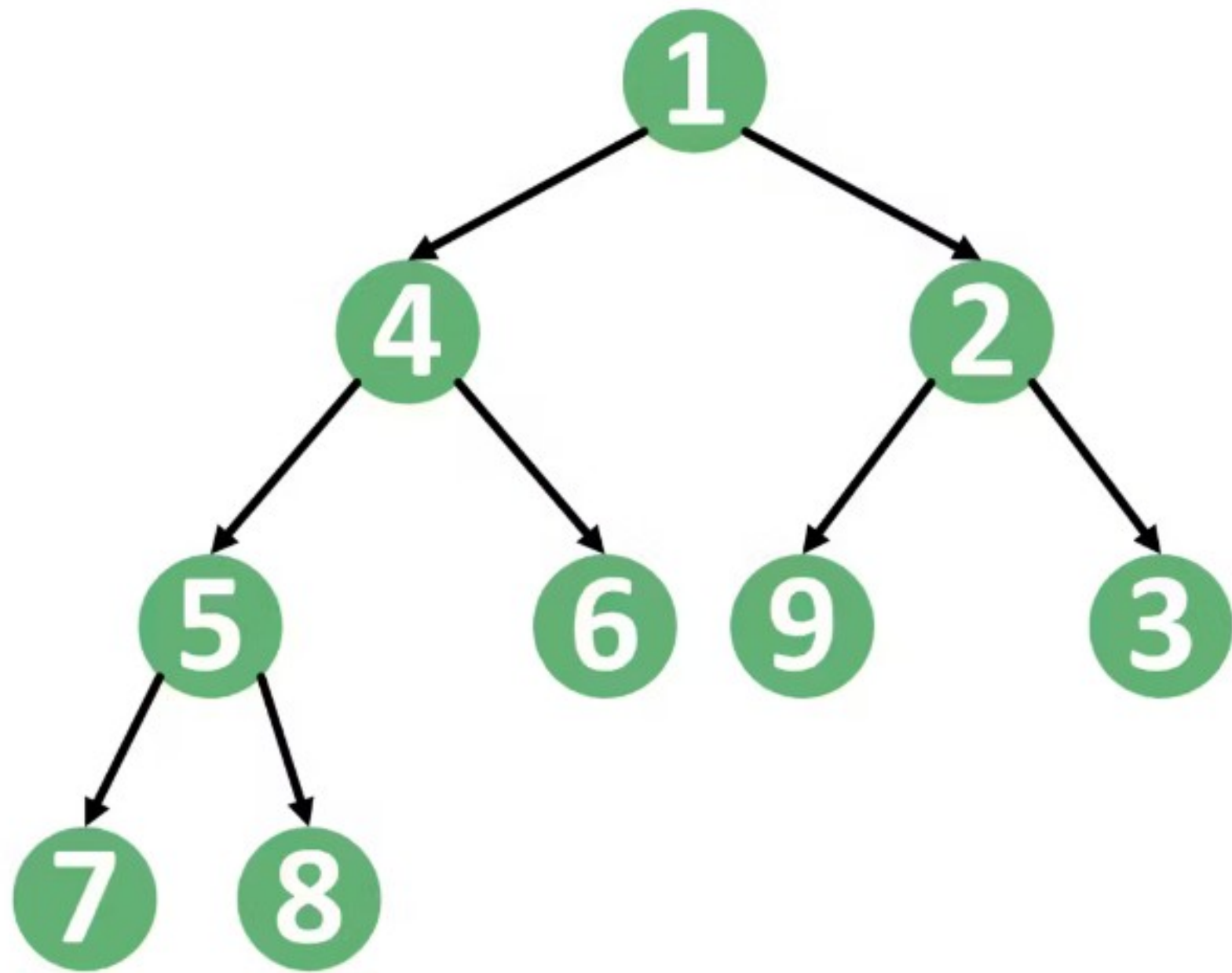
Er dette en Max Heap?



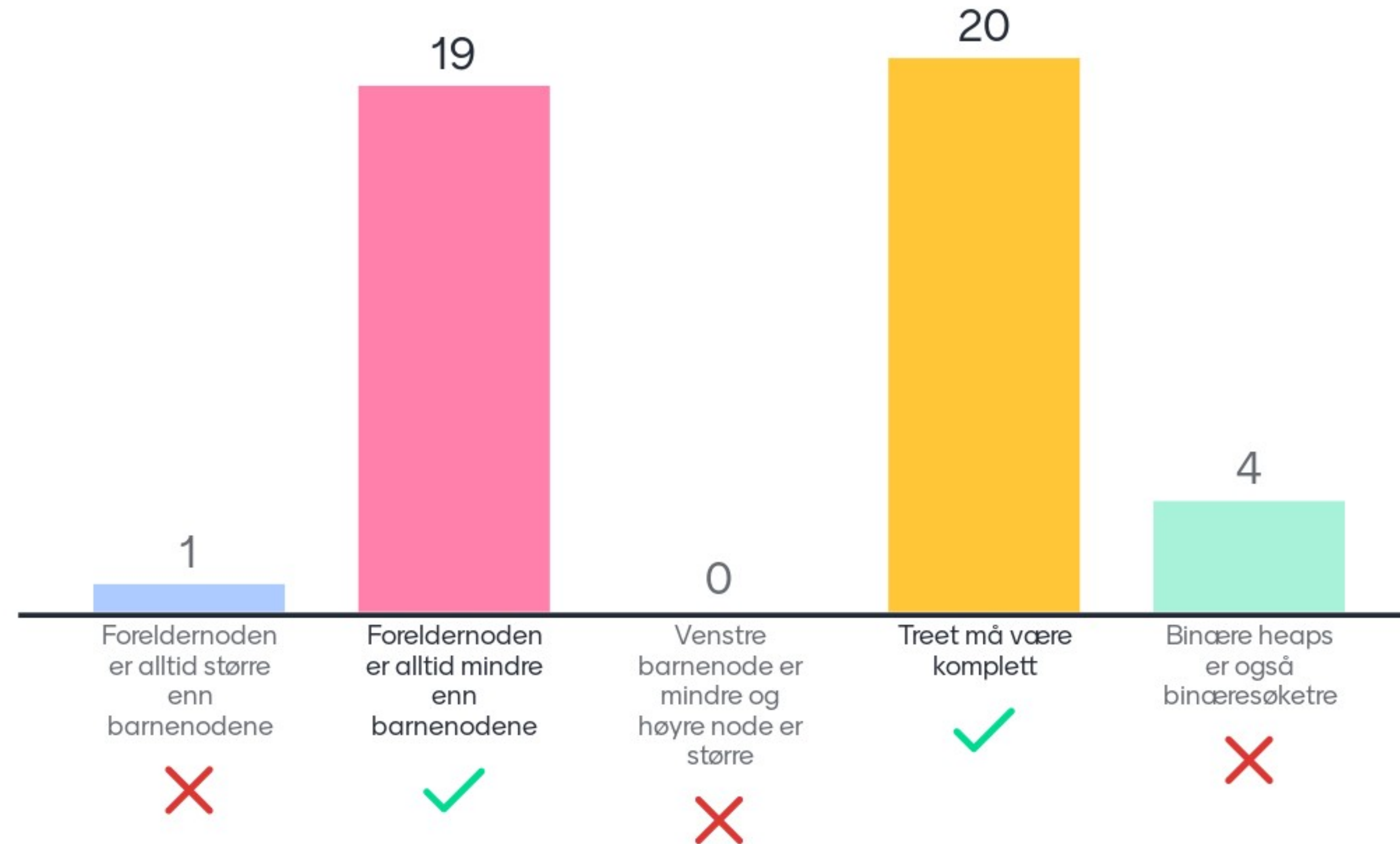
Er dette en Max Heap?



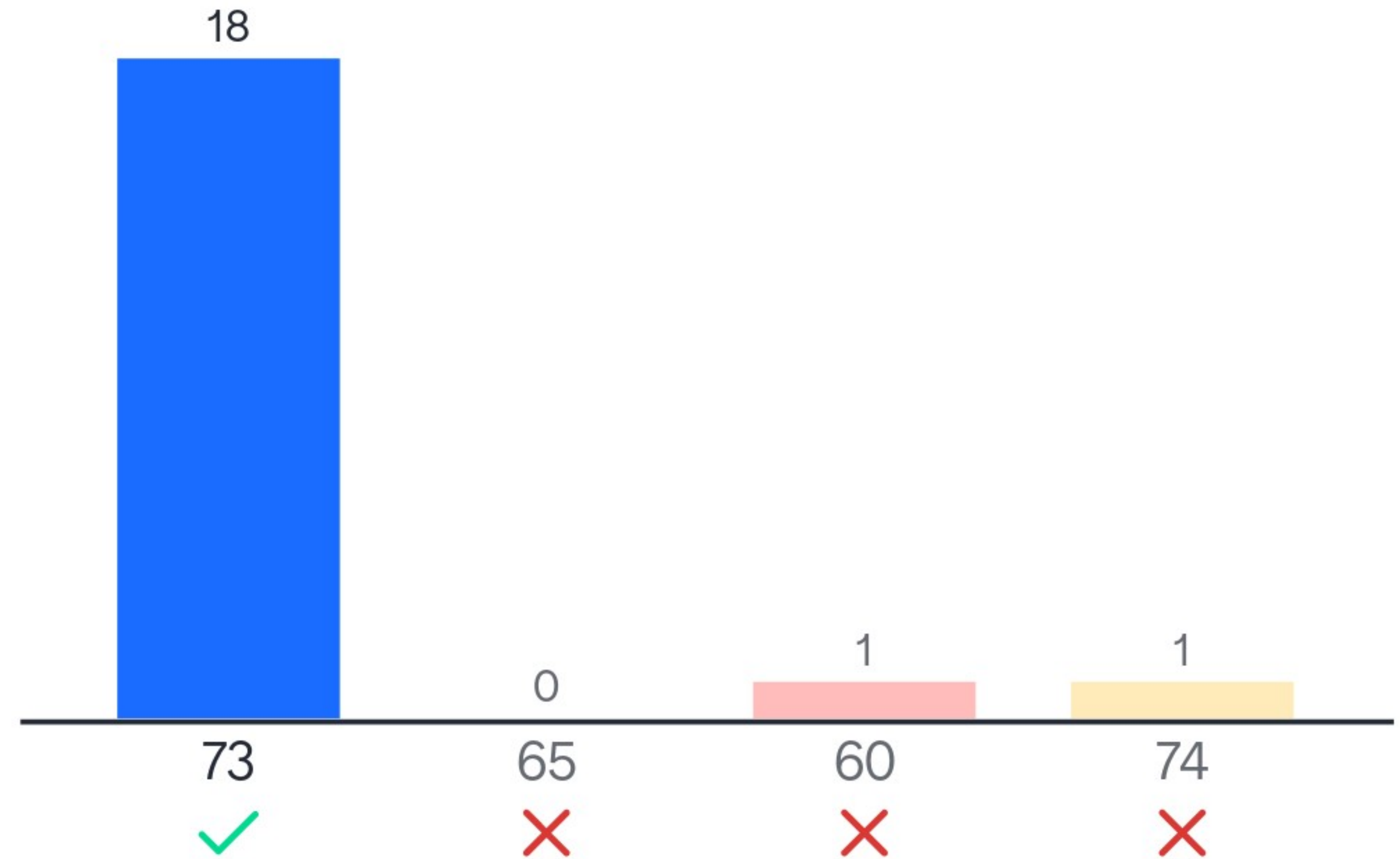
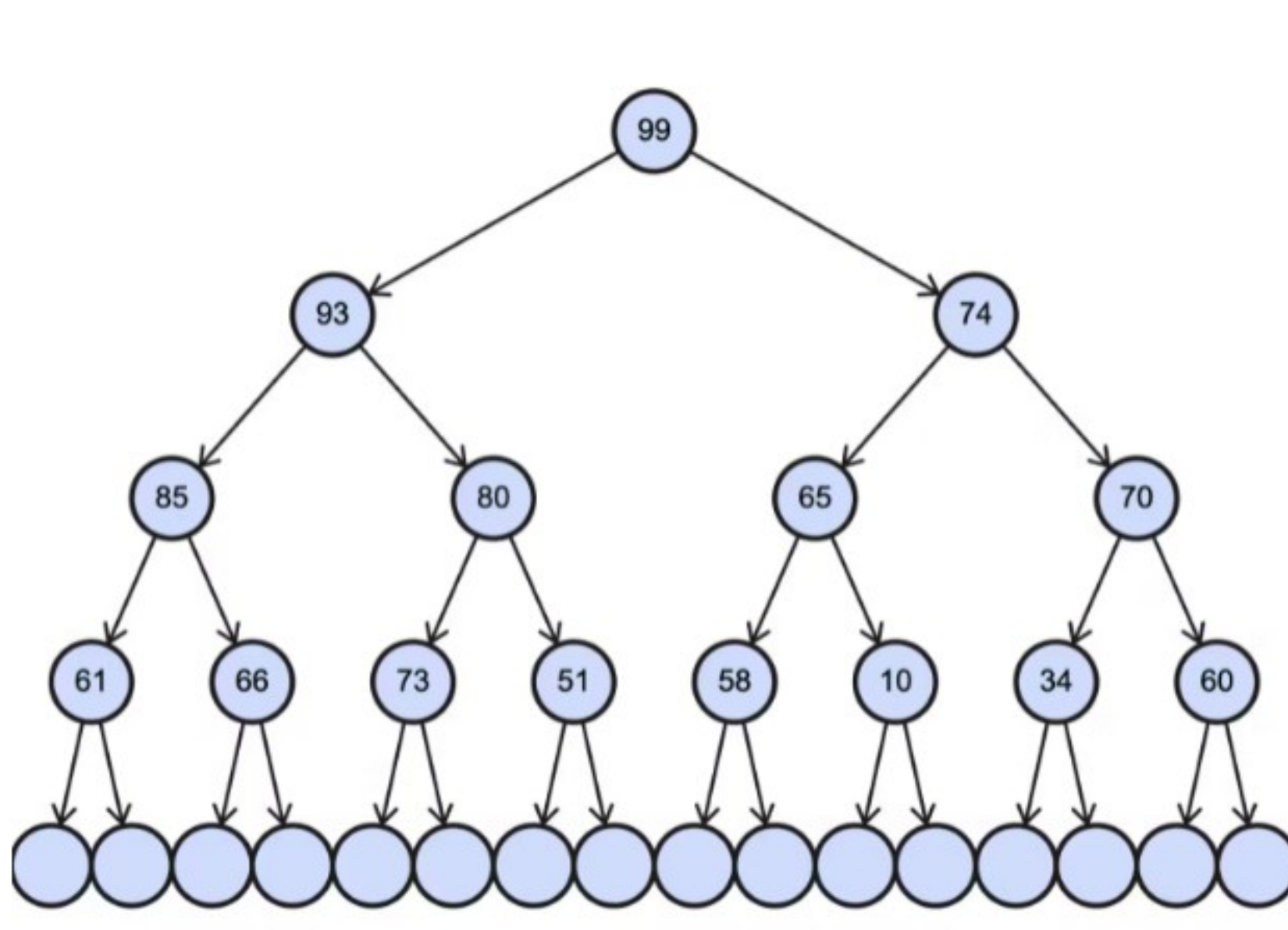
Hva er indexen til foreldernoden til
"6"

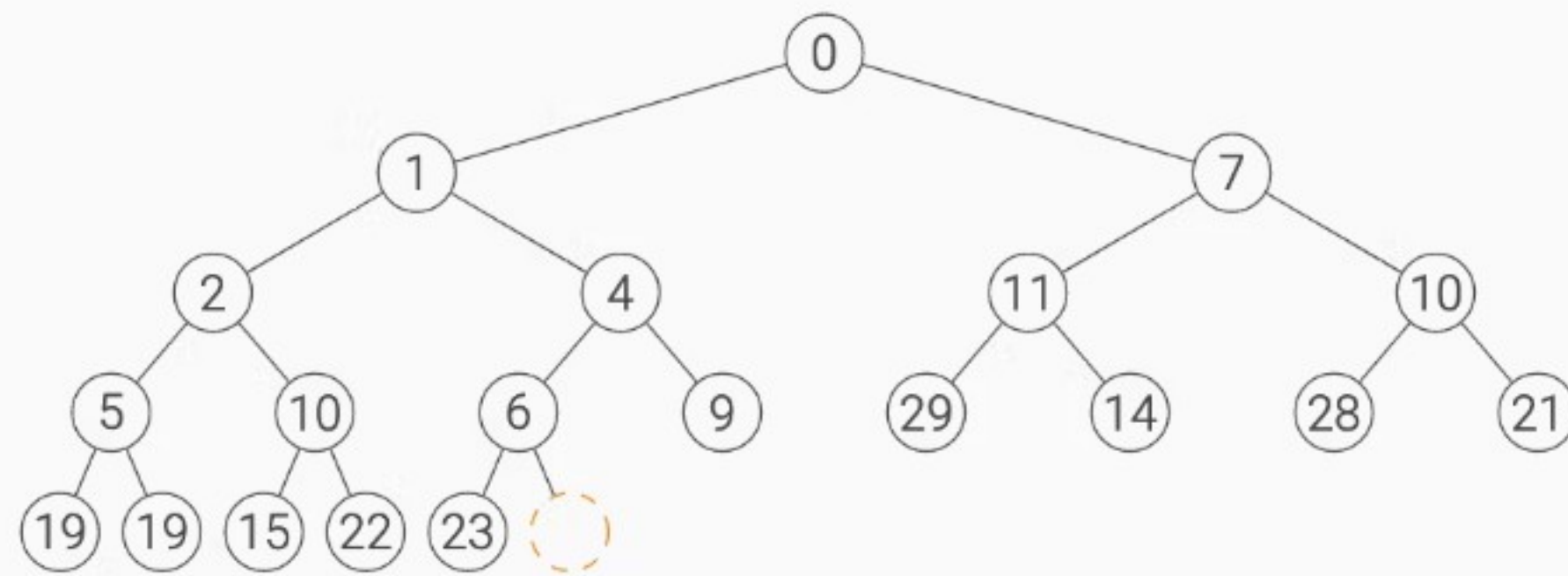


Hvilke egenskaper er sann for en min heap?



Hva er den største verdien en node på siste rad kan ha?





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)

Fellesoppgaver

Oppgave 1 Prioritetskø (vekt 10%)

En *max heap* er en binær heap hvor ordningskravet er slik at barn alltid er *mindre eller lik* sine foreldre.

Gitt verdiene 13, 24, 5, 6, 18, 57, 9, 12, 35, 8, 16, 3, 20, 35 og 14

1a Innsetting (vekt 5%)

Vis array-representasjonen du får ved å sette inn verdiene en om gangen i en initielt tom max heap.

1b Lineær tid-oppbygging av heap (vekt 5%)

La B være array-representasjonen av et komplett binært tre med verdiene gitt over. Vis array-representasjonen av *max heapen* du får ved å bruke lineær tid *buildHeap*-algoritmen på B .

Eksamen 2011(Bare 1b)

Heap (4 poeng)

Gitt følgende array (index 0 er tom). Representerer dette arrayet en heap? Begrunn svaret.

[, 9, 5, 7, 8, 25, 13, 9, 6]

Skriv ditt svar her...

Eksamen 2018

Eksamen 2022

Kan et AVL brukes som en prioritetskø med samme kjøretid som en binærheap?

Hvorfor/Hvorfor ikke?

Kan et AVL brukes som en prioritetskø med samme kjøretid som en binærheap? Hvorfor/Hvorfor ikke?

som faen

Ja(?) uthenting av minste
element i en minHeap er $\log(n)$,
samme vil vel være for AVL

jajajajajaja

jonas vil fortelle hvorfor

yes, begge har samme
kjøretid siden de er
balanserte

kommer ann på hva O
notasjon for balanse er

Huffman-koding

Hva er vanskelig med huffman-koding?

alt

vite når man skal bruke det.

dekode et huffman tre

hvor man skal plassere
noder

Når man skal tegne det, har
"rekkefølgen" noe å si. Om alt
blir plassert på samme nivå,
eller når man skal bevege seg
"oppover".

rekkefølge på kombinasjon
av noder

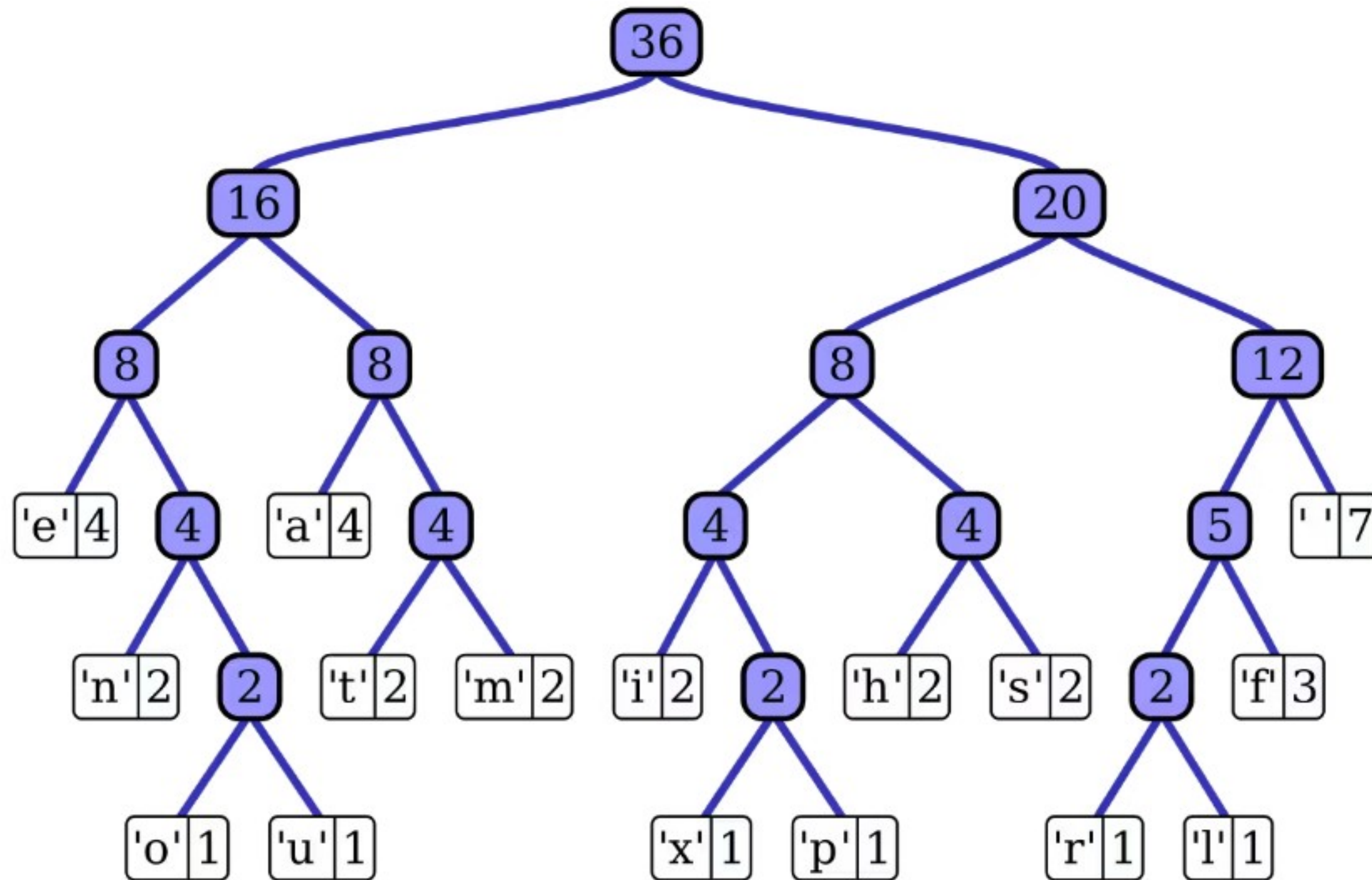
Finne frekvensen til hvert
symbol

er huffmantrær komplette?

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

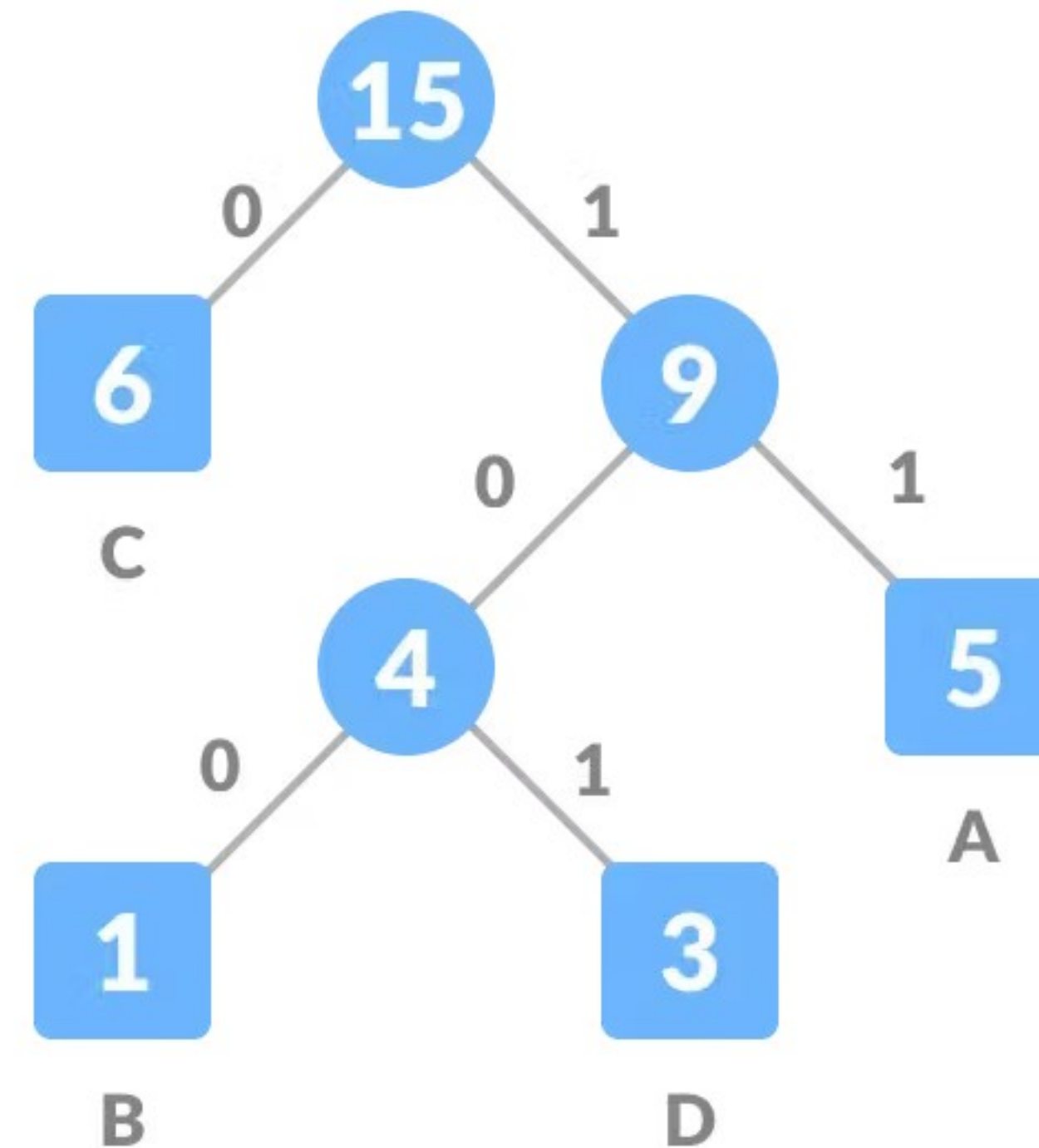
- Huffman træ er treet som viser frekvensene av symboler
- Start med å lage en frekvenstabell
- Lag en prioritetskø som prioriterer basert på frekvens
- Velg de to minste elementene fra køen
- Lag en ny node, som har de to nodene som bar
- Legg den nye noden tilbake i køen

Om Huffmantrær

- Hvis vi lager et huffman-tre med n antall symboler så vil høyden på treet være maksimalt $n-1$
- Et huffman-tre med n antall symboler vil ha $2n-1$ antall noder

Bestemme huffman kode

- Se på linjene(stien) fra rotnoden til den gjeldende noden
- Hver høyre: 1
- Hver venstre: 0



Demonstrasjon

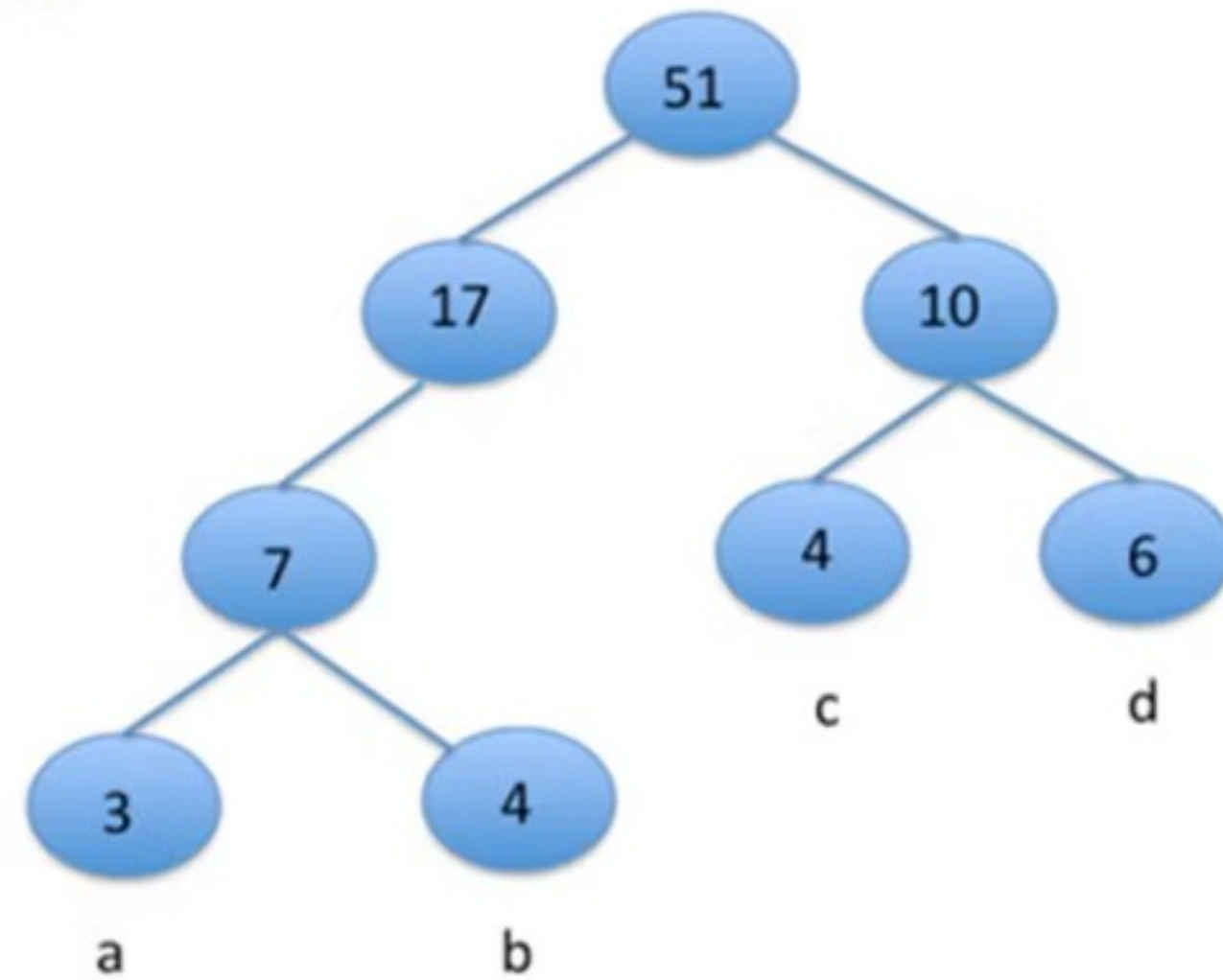
Fellesoppgaver

Oppgave 3a: Huffman

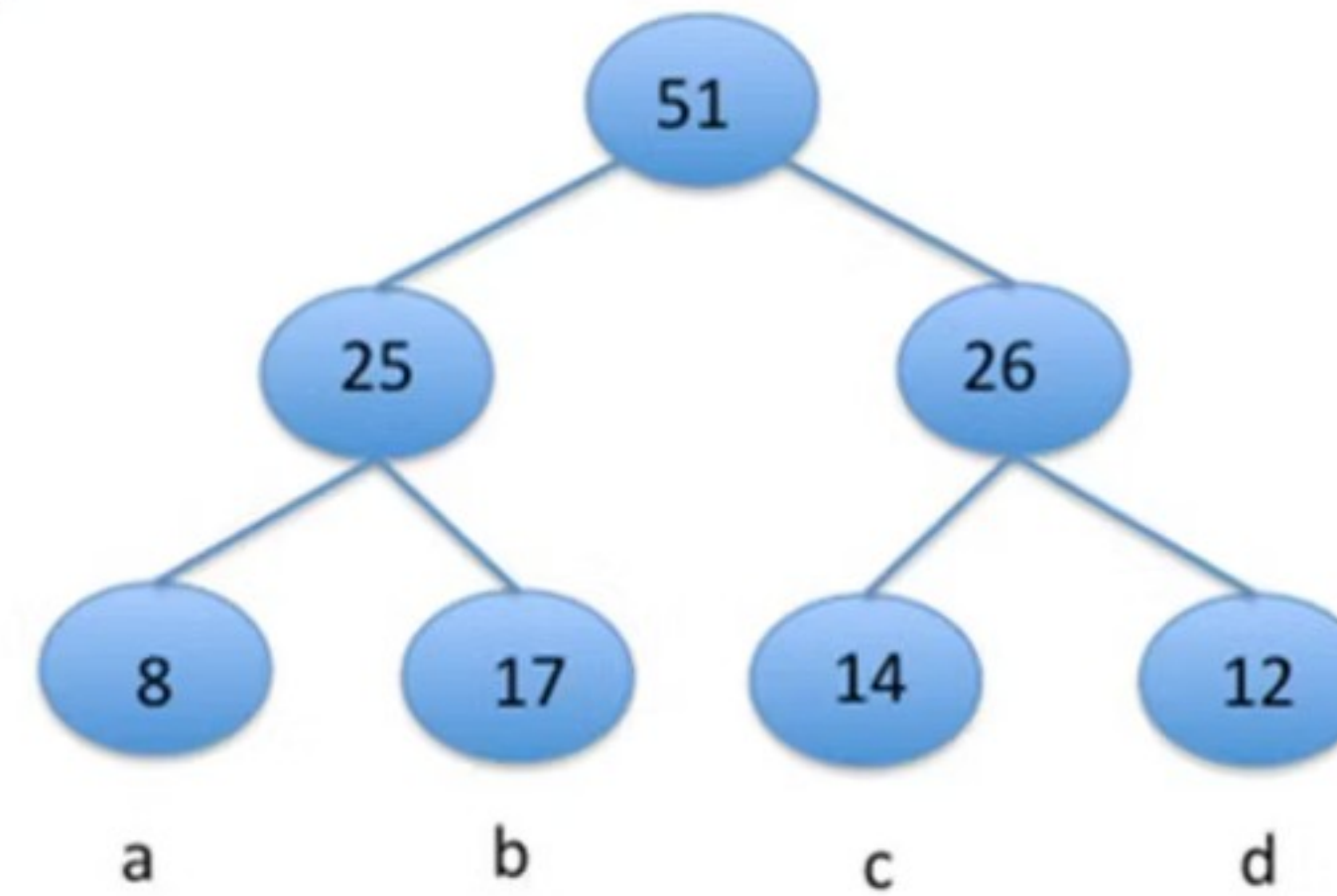
Gitt de to trærne A og B under. For hvert tre, er treet et gyldig Huffman-tre? Hvis det ikke er gyldig, forklar hvorfor og endre det slik at det blir gyldig. Du kan anta at vektingen til løvnodene er riktig.

I denne oppgaven skal du svare med digital håndtegning. Bruk eget skisseark (utdelt). Se instruksjon for utfylling av skisseark på pult.

A:



B:



Eksamen 2017

Oppgave 3 Huffmankoding (vekt 10%)

Anta at en input-fil har gitt deg følgende frekvenstabell:

- a: 9
- b: 2
- c: 4
- d: 2
- e: 1
- f: 1

3a Huffmantre (vekt 10%)

Tegn et Huffmantre basert på frekvenstabellen.

Eksamen 2010

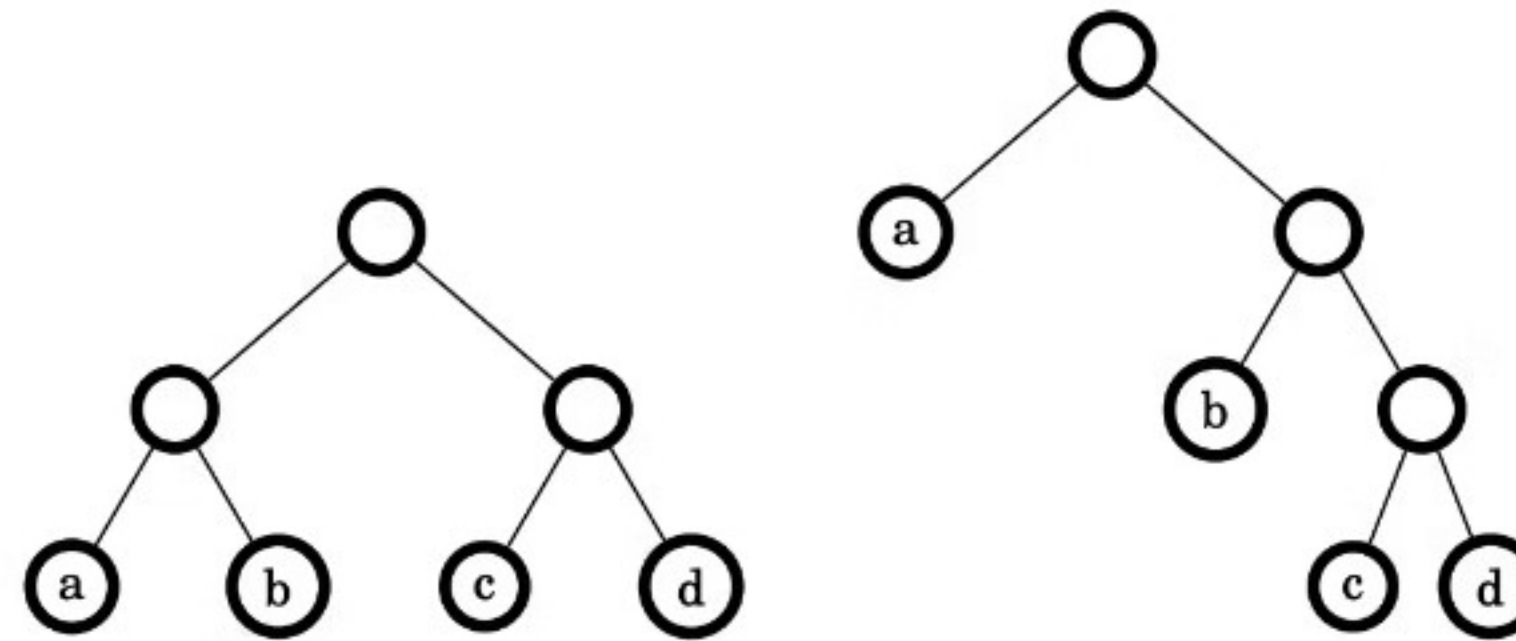
2b Huffman-koding (vekt 5%)

Gitt et alfabet (= reservoar av tegn) med størrelse 16 (la oss si a, b, \dots, p), og en tekst med lengde 100 tegn. Anta en Huffman-koding hvor tegnet a er kodet med én bitt, for eksempel 0. Hva kan du slutte om frekvensen av tegnet a i teksten?

2c Huffman-koding (vekt 5%)

Vanligvis kan en gitt tekst ha mer enn én Huffman-koding som er optimal som definert i forelesningen. Anta en tekst som bruker et alfabet med 4 tegn: a, b, c , og d .

Spørsmål: Er det mulig at teksten har 2 forskjellige Huffman-kodinger som gitt i treet til høyre resp. til venstre?



Hvis ditt svar er *Ja*, vis dette med et eksempel, dvs. en frekvenstabell for de 4 tegnene. Hvis ditt svar er *Nei*, begrunn svaret.

Eksamen 2012

Gruppeoppgaver