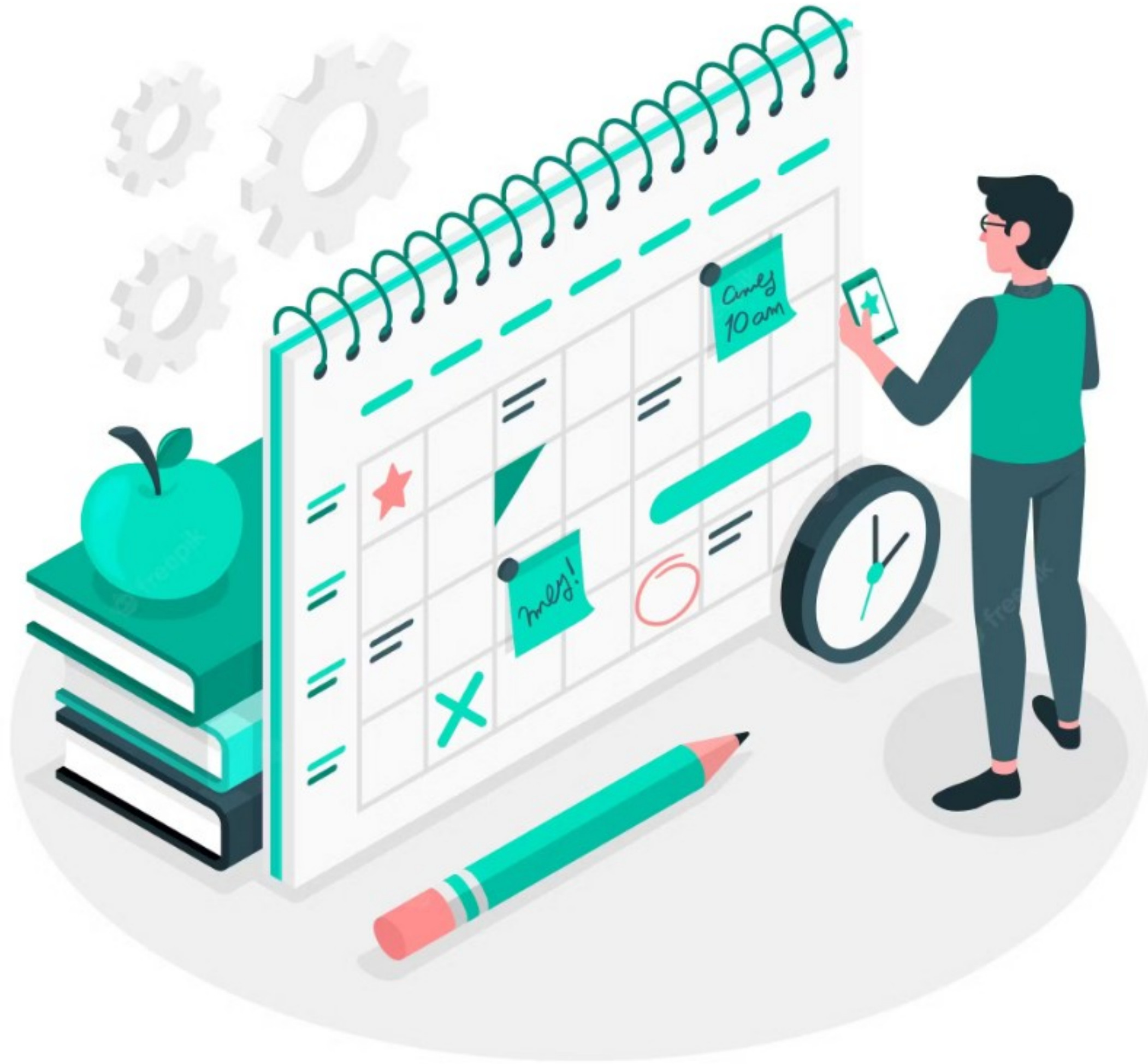


IN2010 Gruppe 4

Uke 10 - Hashing



Dagens Plan

- Info
- Pensumgjennomgang
- Gruppeoppgaver

IN5060 - User studies

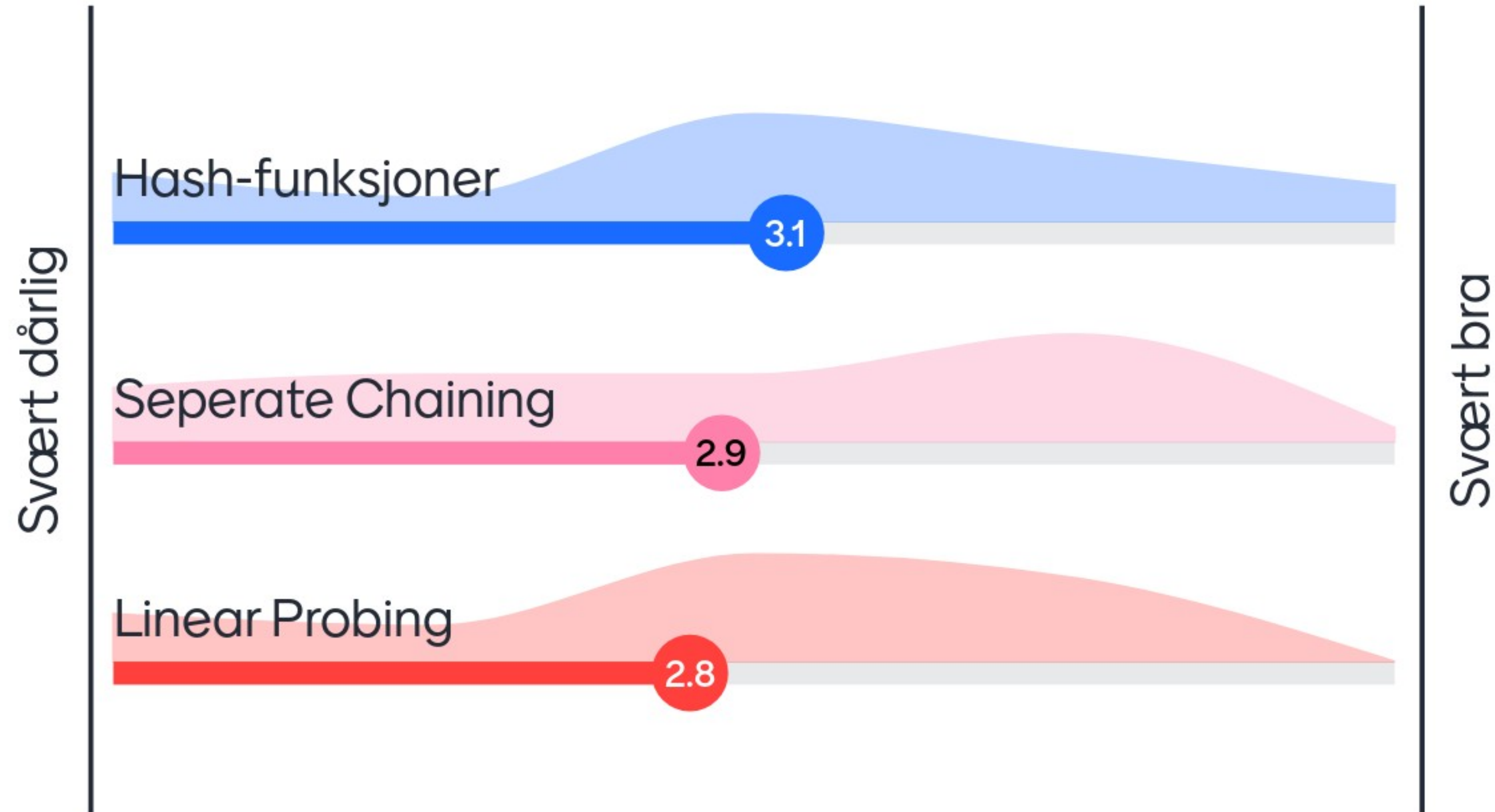
<https://forms.gle/ARWQ9uAoYAcPZzri8>

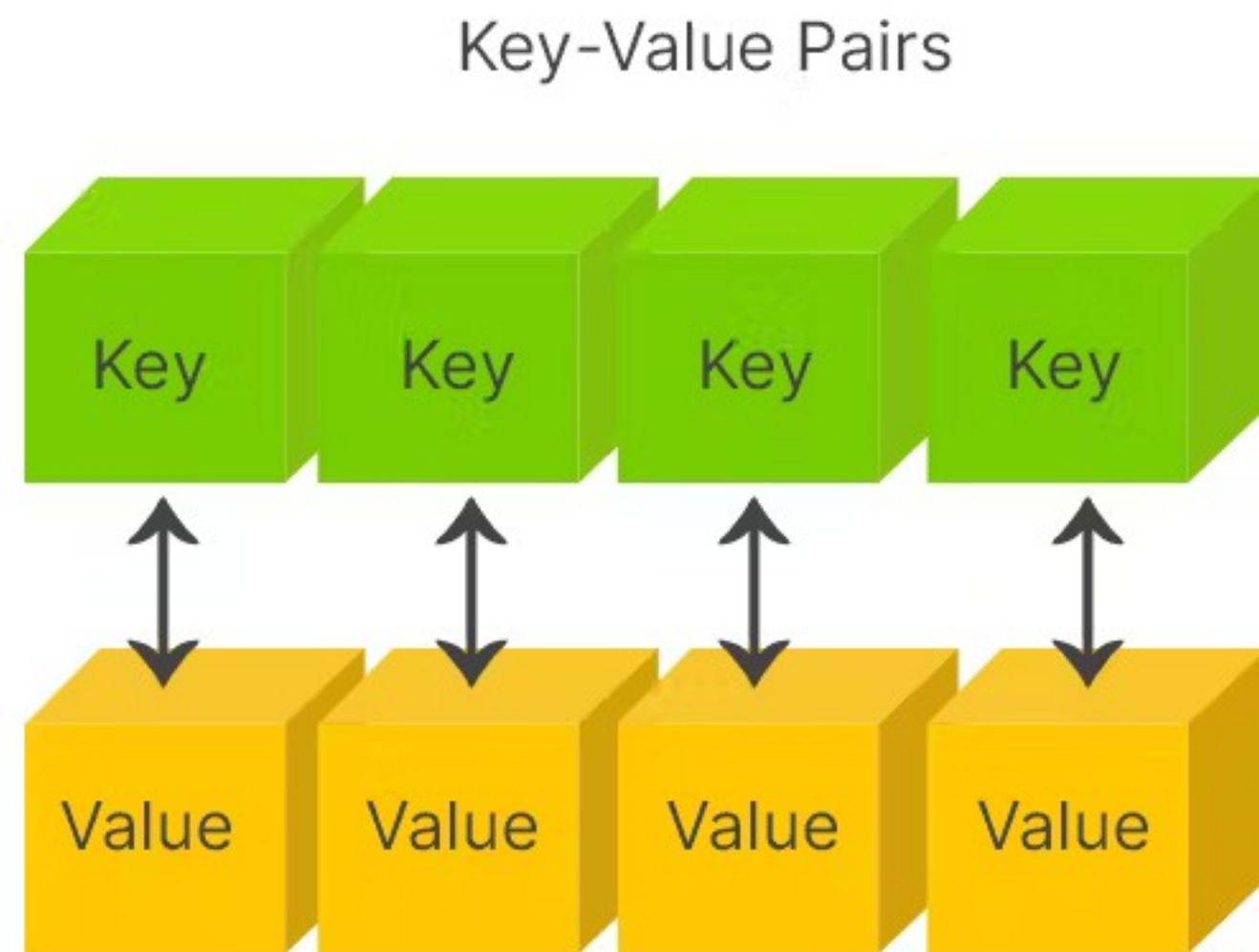
Innlevering 4 er ute!

<https://www.uio.no/studier/emner/matnat/ifi/IN2010/h23/innleveringer/innlevering4.pdf>

Pensumgjennomgang

Hvor godt forsto du ukens pensum?



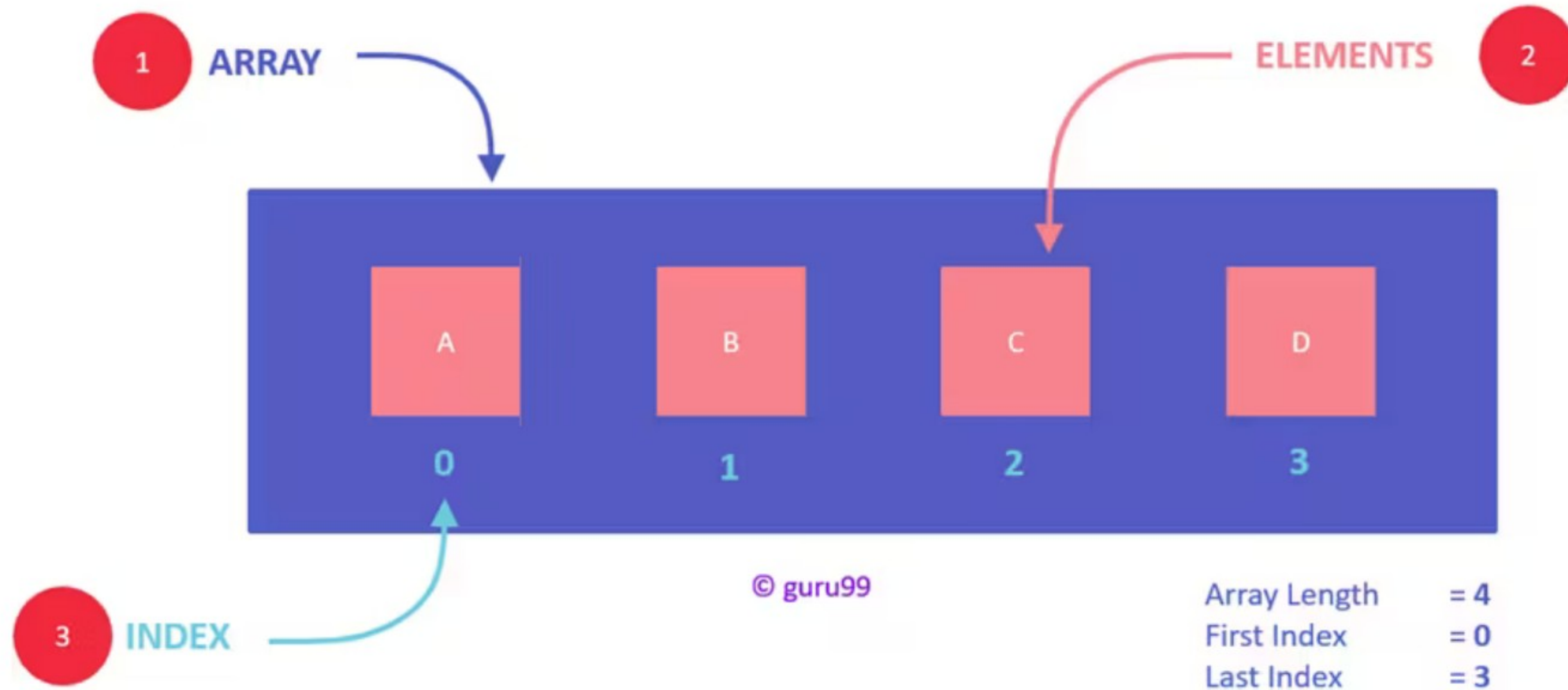


Map

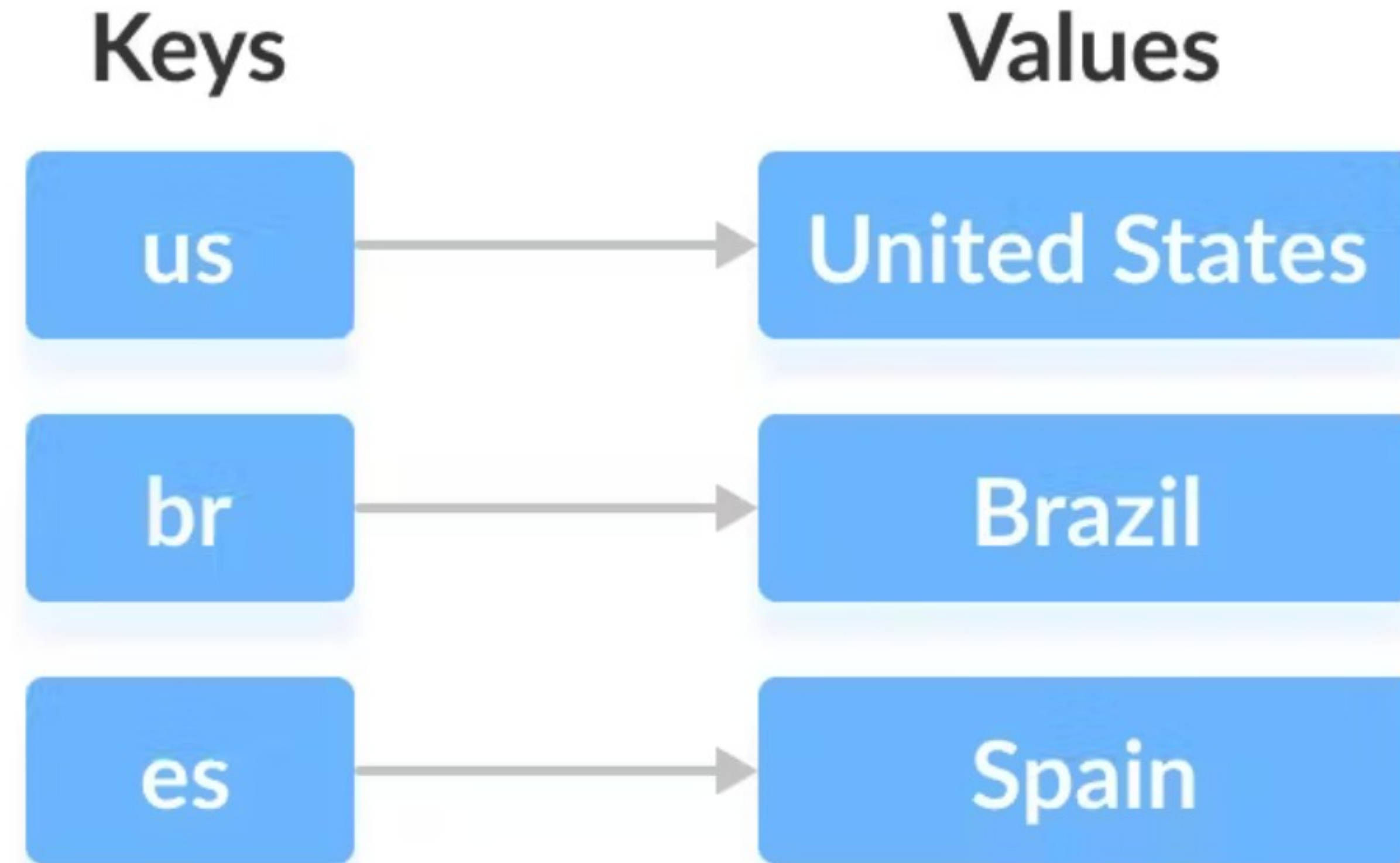
- Gir en nøkkel e hvis verdi
- Oppsøker du en nøkel, så får du verdien
- Eksempel: Dictionaries i python

Eksempler på Maps

CONCEPT DIAGRAM



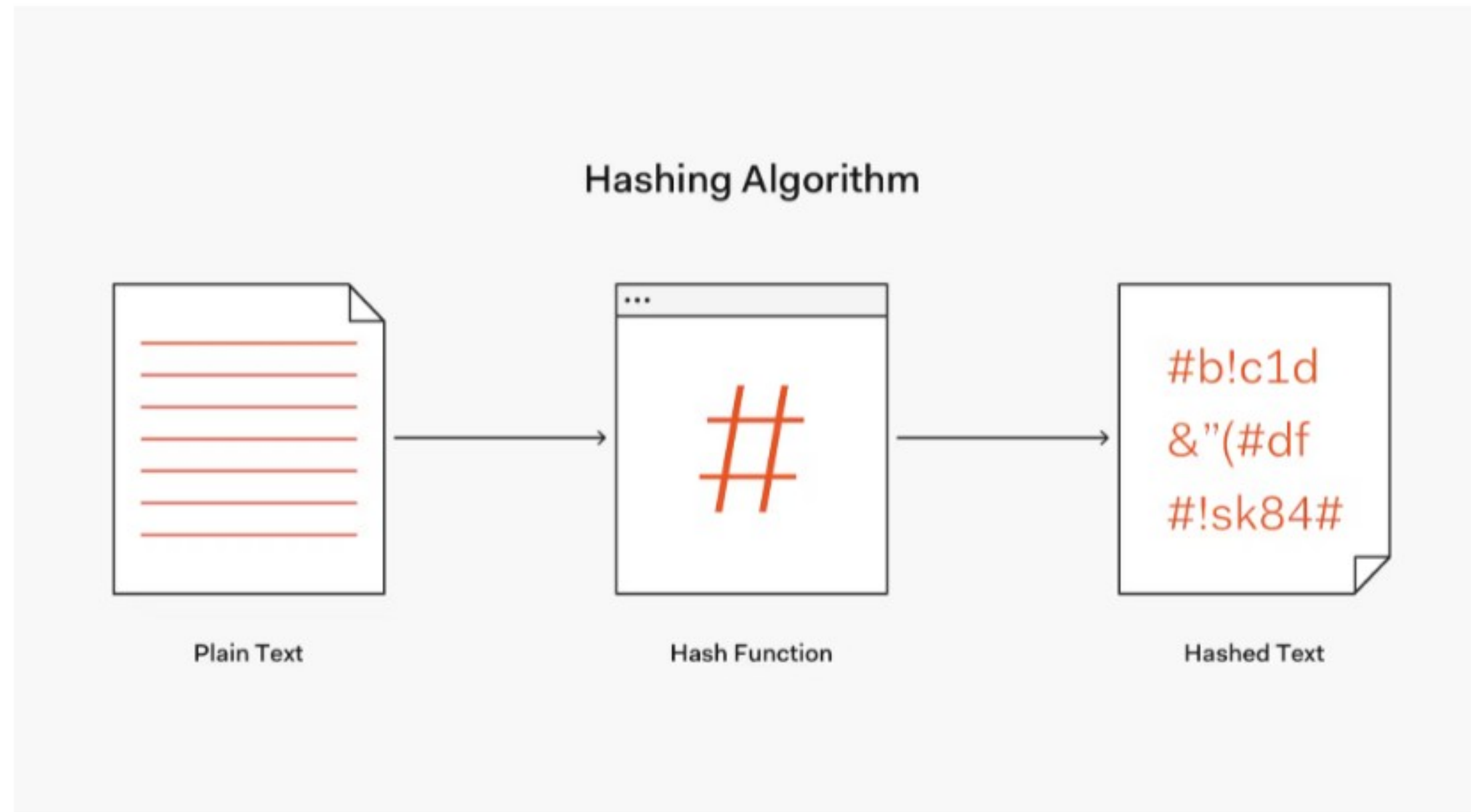
Arrays



Dictionaries

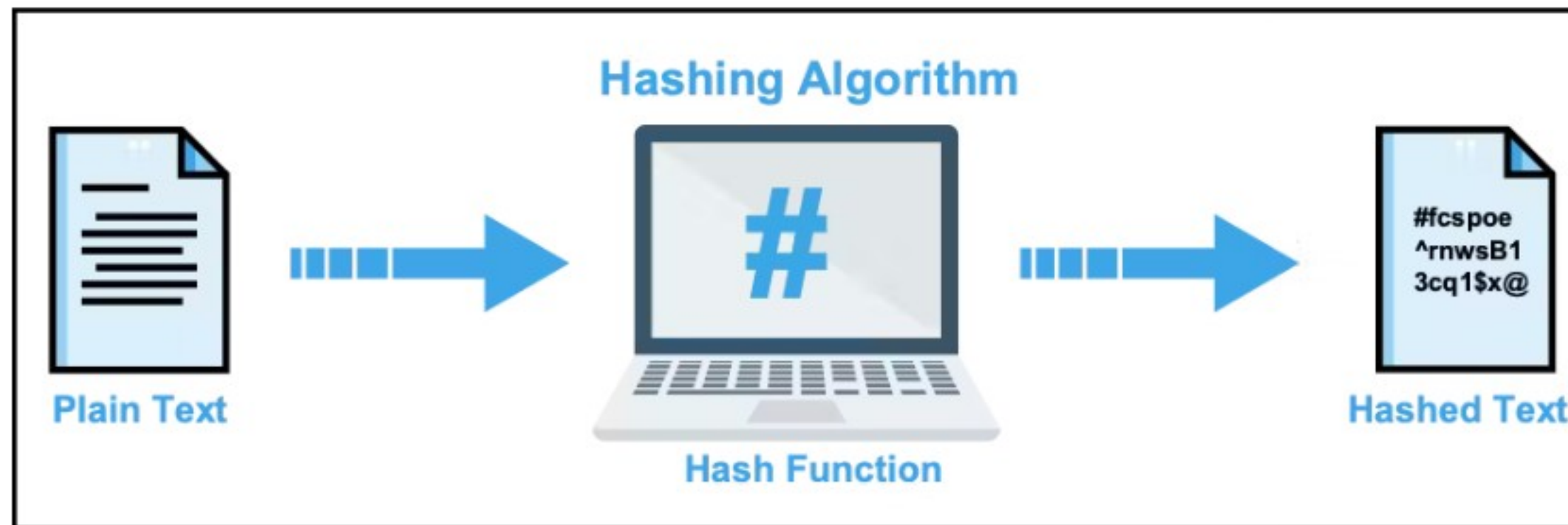
Problem?

- Nøkler er lagret i sin originale form
- De med nøklen kan få tilgang til verdien/dataen
- Generelt veldig usikkert for store datamengder



Hashing

- Ideen av hashing er å ta en Nøkkel/Verdi(steng, tall, etc)
- Opprette en funksjon(hashfunksjon) som endrer denne verdien



Hash-Funksjoner

- En funksjon som tar inn en gitt input og returnerer samme output hver gang
- Må også sørge at ulike inputs får ulike outputs(Så langt det lar seg gjøre)
- Eksempel: Input "Hei" skal gi samme output for hver gang



Hash-funksjoner er endeløse!

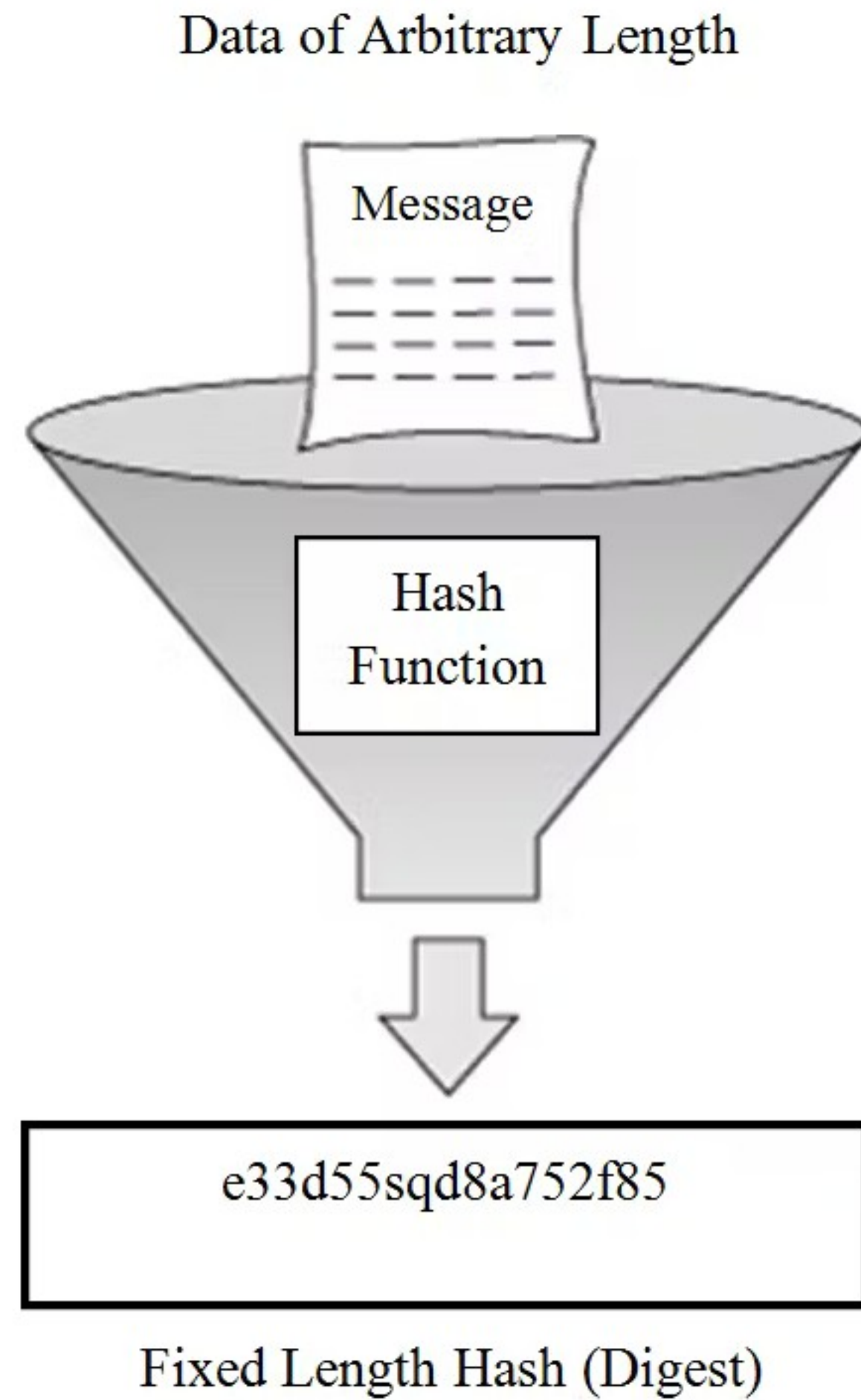
- Binærsøk på en streng
- Dobble strengen
- Reversere strengen
- Modulo of et tall

ALGORITHM: EN LITT DÅRLIG HASHFUNKSJON PÅ STRENGER

Input: En streng s og et positivt heltall N **Output:** Et heltall h slik at $0 \leq h < N$

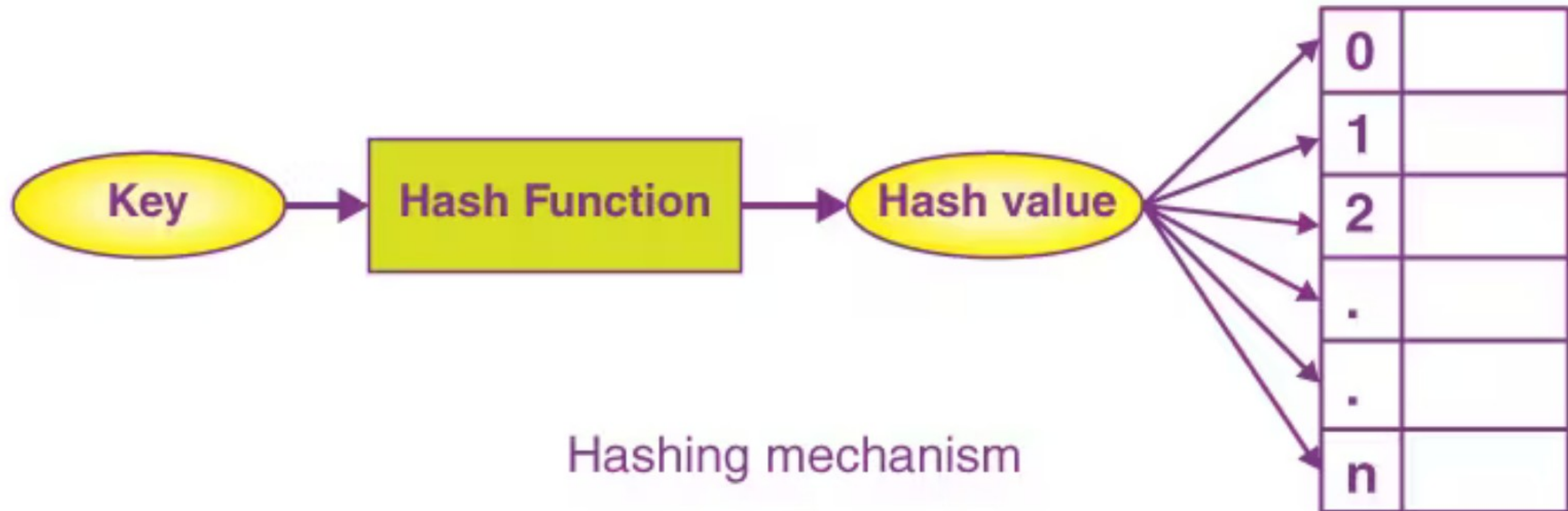
```
1 Procedure HashStringBad( $s, N$ )  
2    $h \leftarrow 0$   
3   for every letter  $c$  in  $s$  do  
4      $h \leftarrow h + \text{charToInt}(c)$   
5   return  $h \bmod N$ 
```

Eksempel fra forelesning



Hash-Map

- Dette materialiserer datastrukturen MAP
- I dette tilfellet er ikke nøklene "vanlig"
- Men nøkkelen er en hashet verdi fra en annen verdi/tall



Hashing mechanism

Hash-Map

Eksempel

Hash funksjon returnerer lengden på en streng

Hash funksjon problem: Kollisjon

Ulike inputs kan hashes til samme verdi

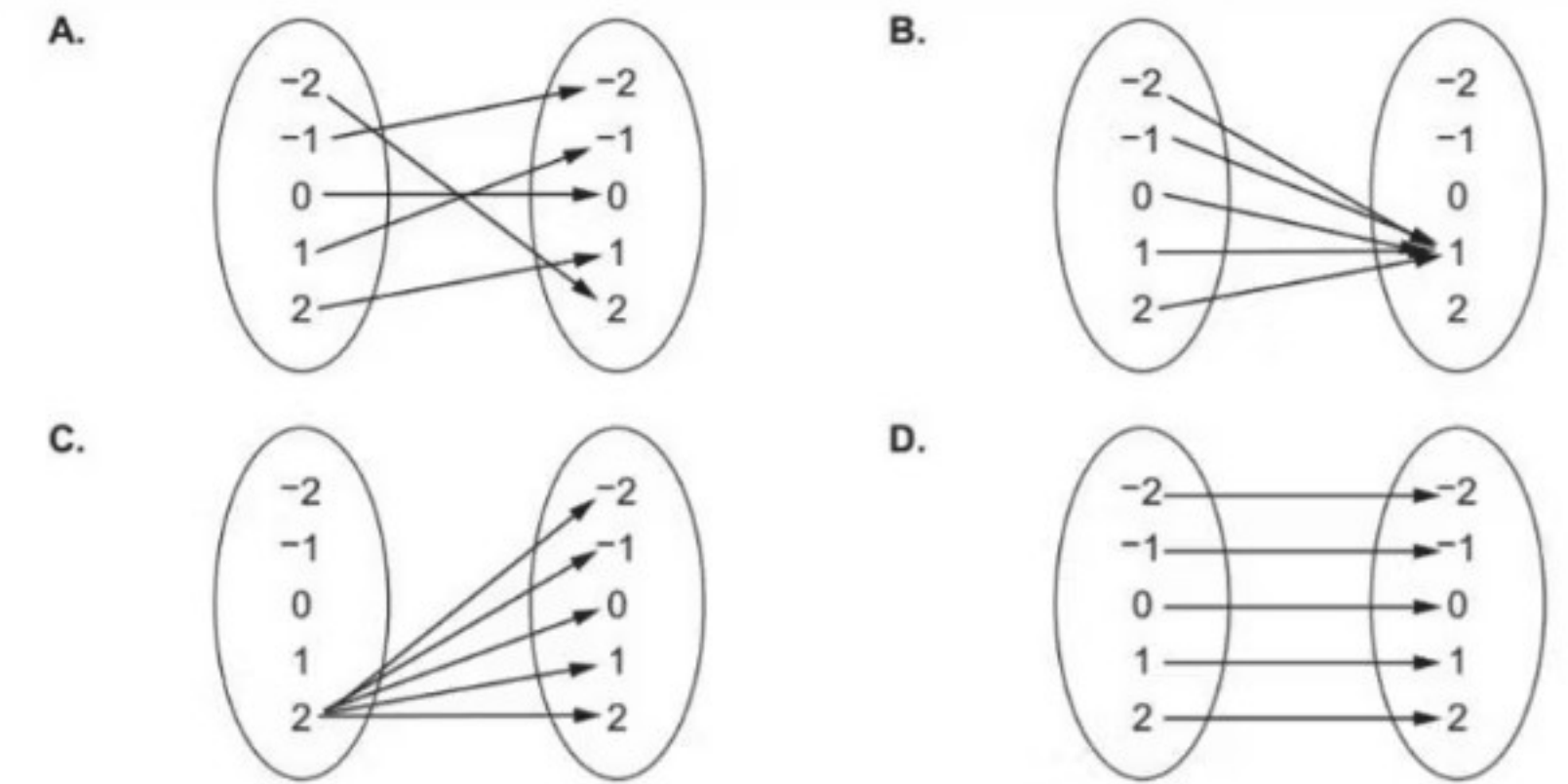
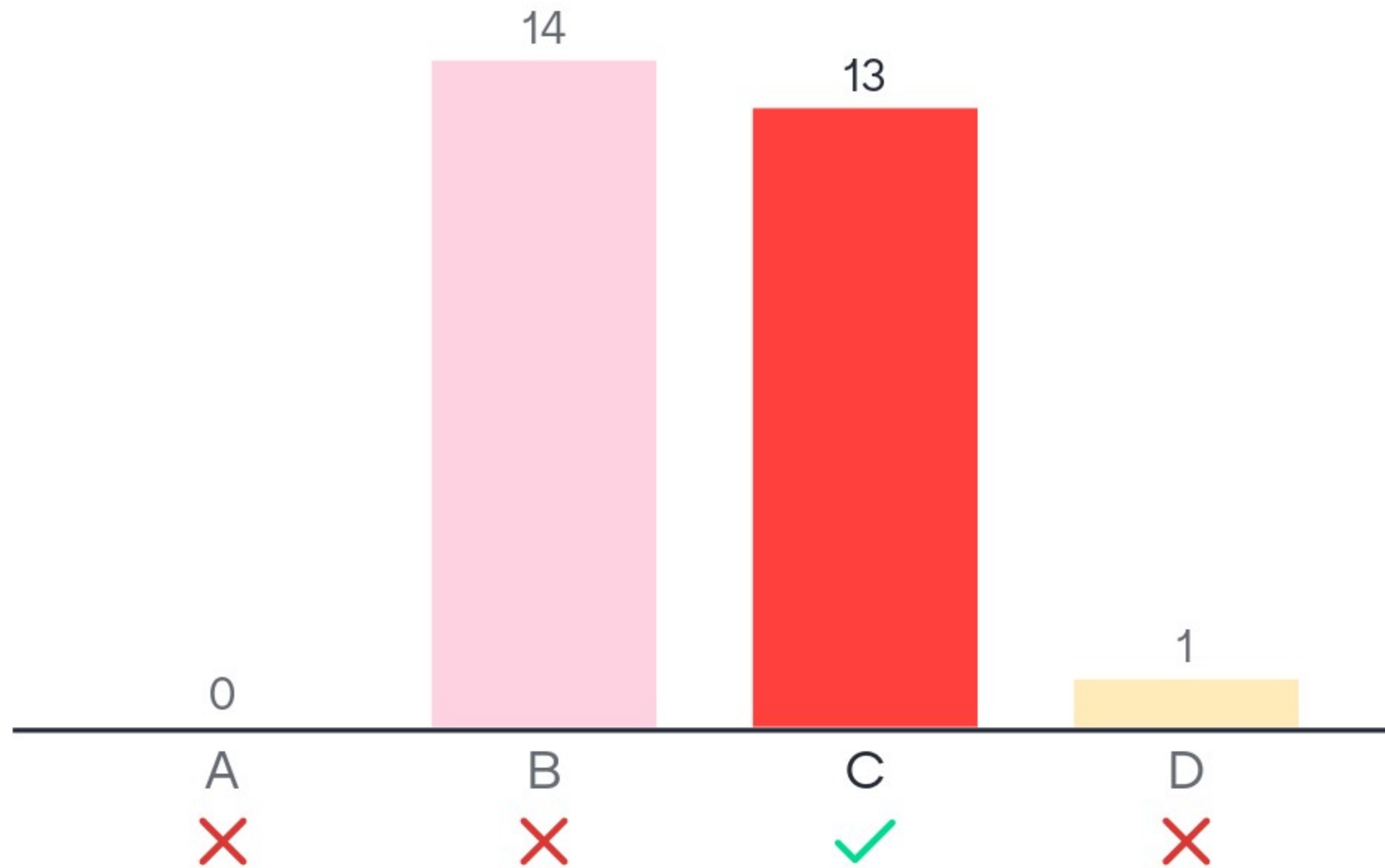
Dette fører til at de "skal" lagres på samme posisjon

Vi introduserer 2 algoritmer for å løse dette problemet

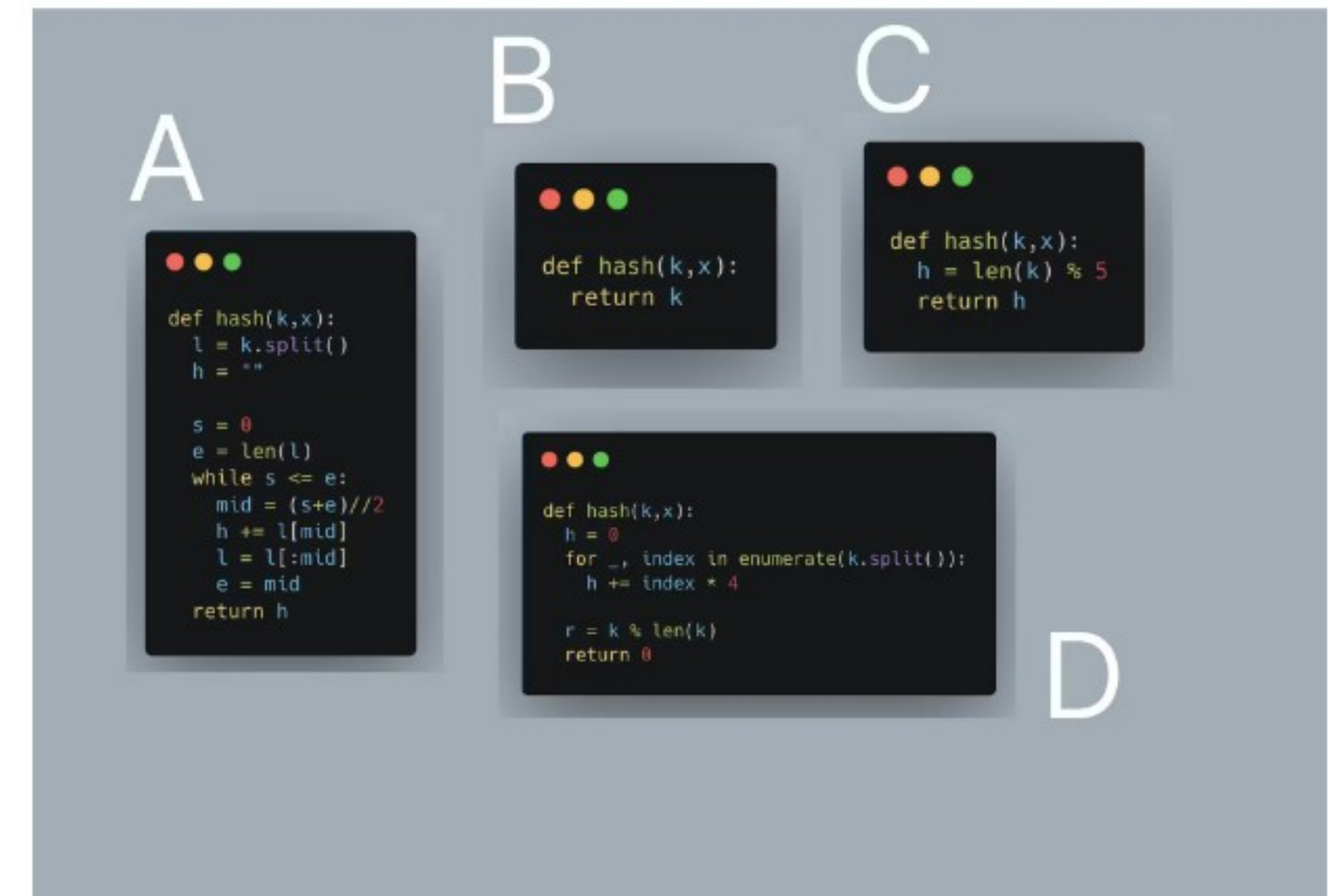
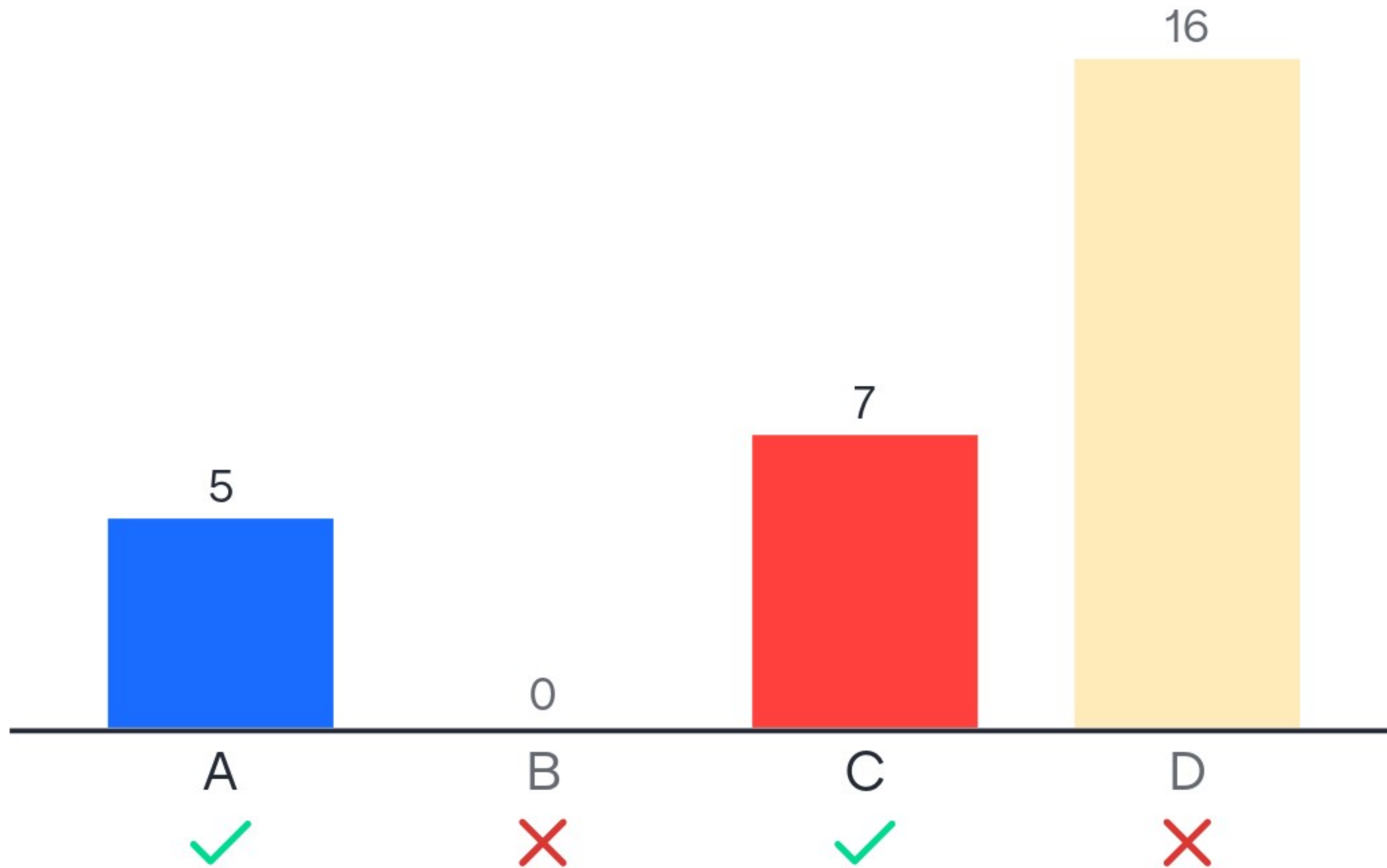
Quiz



Hvilken av disse kan ikke være en hash-funksjon?



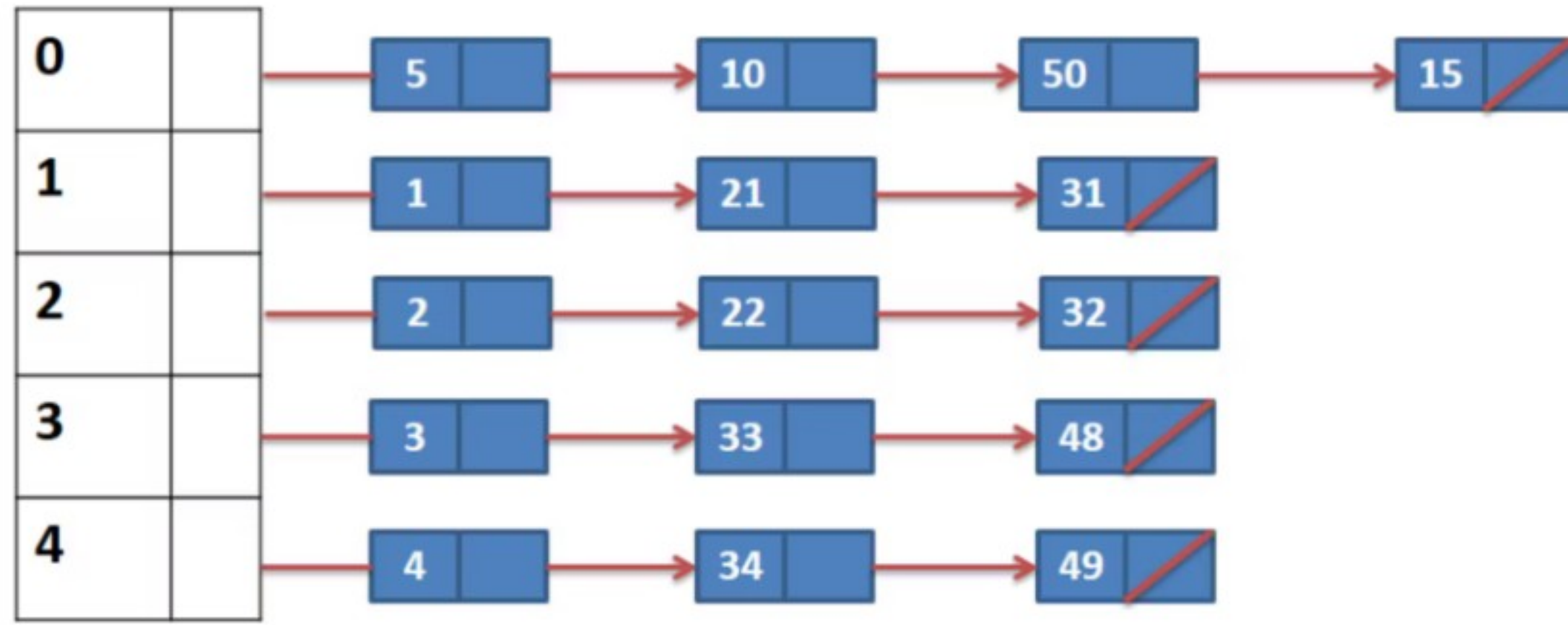
Hvilke hash funksjoner er best?



Kolisjonshåndtering

Kollisjons håndtering - Ide

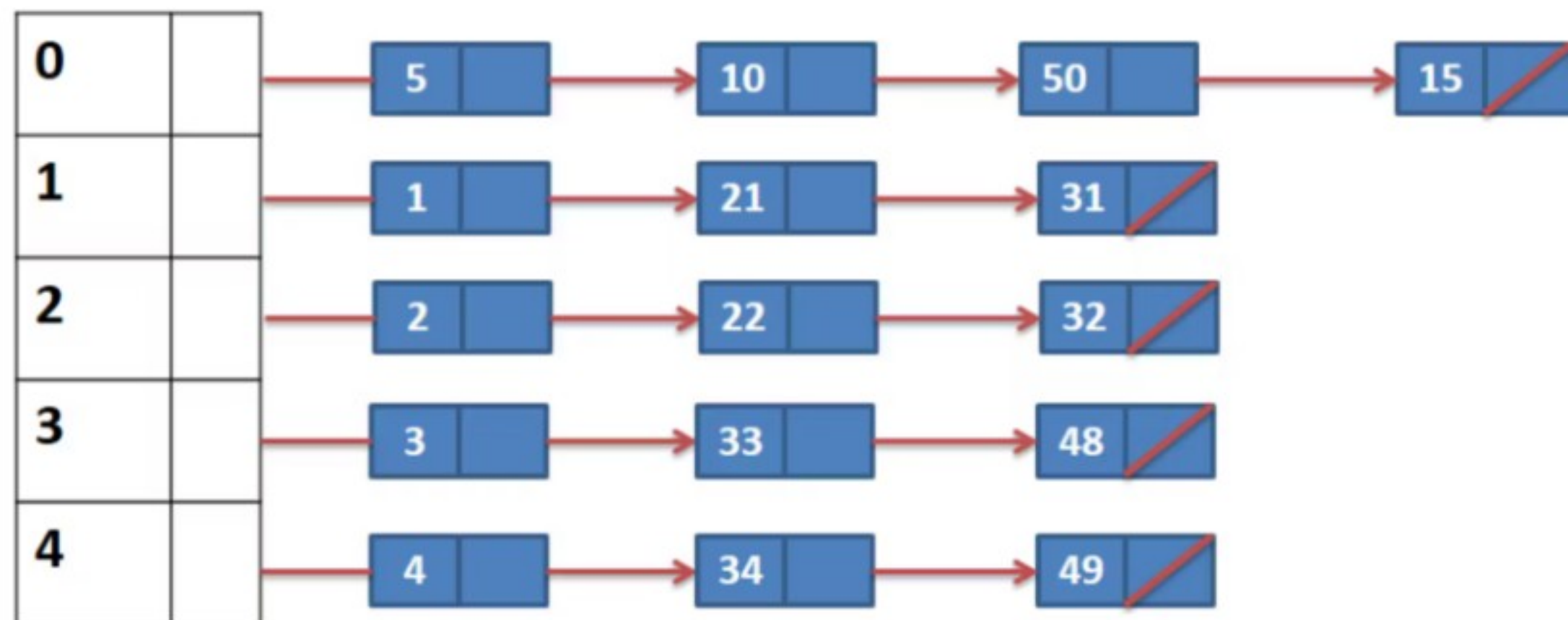
- Håndtere elementer med samme hash verdi
- Sørge for at vi kan finne elementene vi ønsker
- Linear Probing
- Seperate chaining



Hash Table

Seperate Chaining

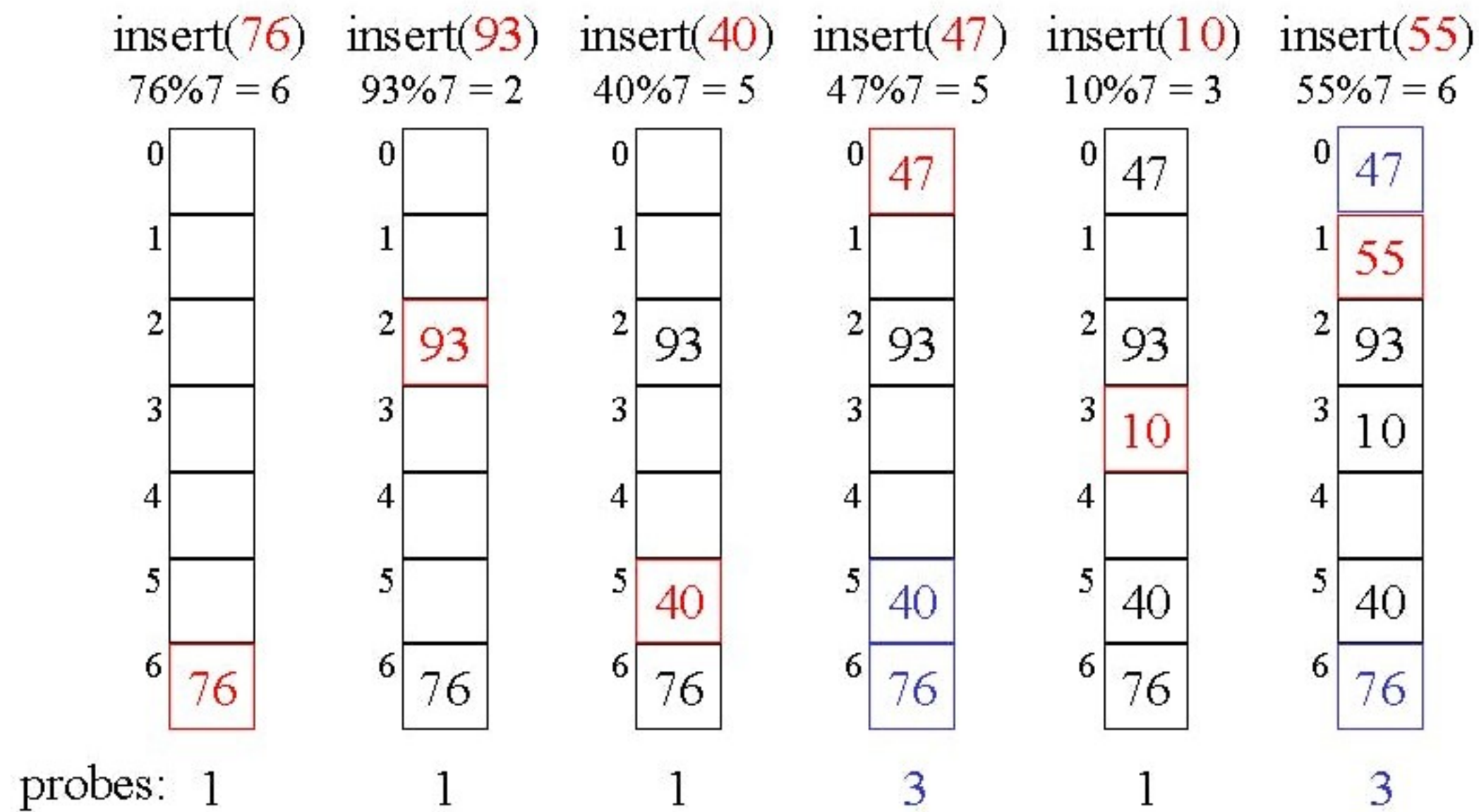
- Ideen her er at nøkler kan hashesh til samme verdi
- Dersom det skjer, så vil elementet legges til en lenkeliste
- Eksempel: Hash funskjon som ser på første sfferet

**Hash Table**

Seperate chaining - Prosedyre

- INNSETTING: Hvis plassen på hashen er tom, lag en liste og sett element der
- Ellers, sett elementet bakerst i lista på hashen
- OPPSLAG: Hvis plassen på hashen er tom, return null
- Ellers, iterer lista i søk om nøkkelen k, og returnner den verdien
- Samme for sletting, bare at vi sletter gitt nøkkel istedenfor å returnere

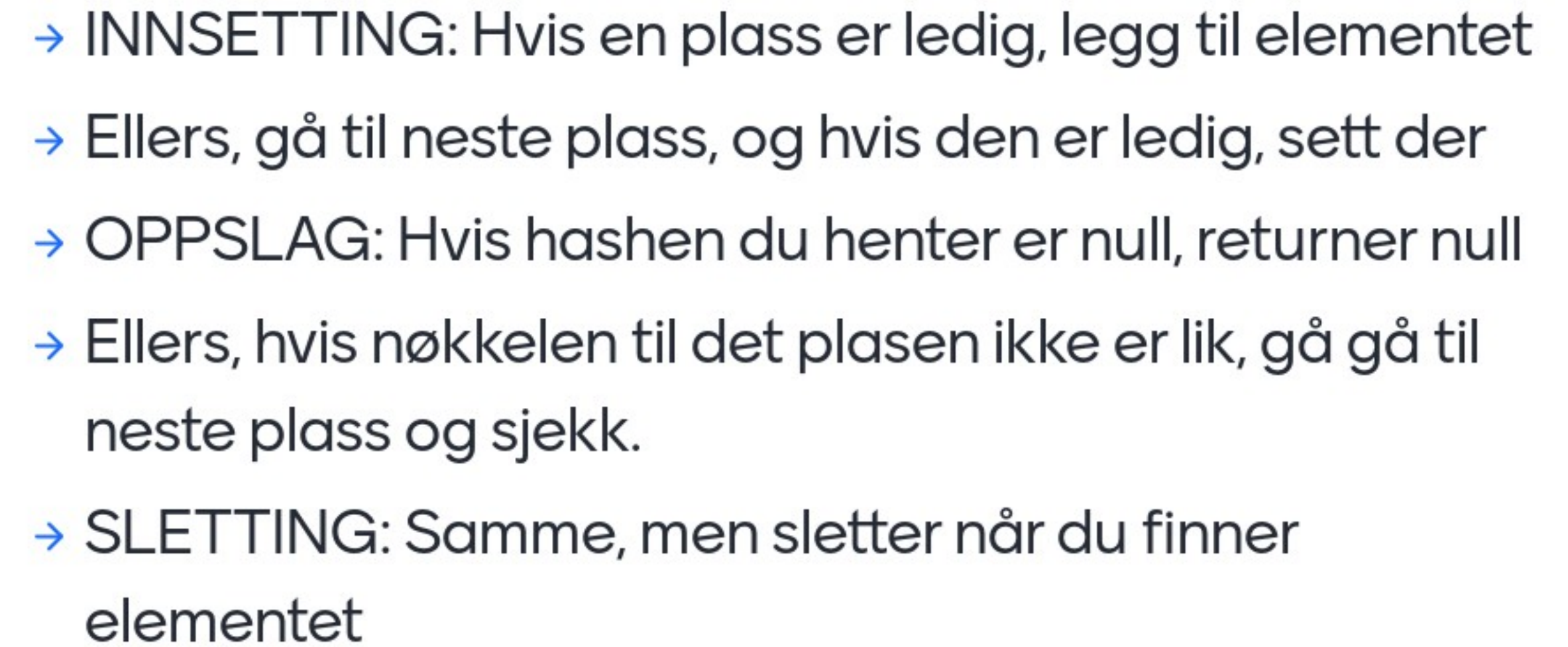
Linear Probing Example



Linear Probing

- Her har vi en hashfunksjon som hasher en verdi til noe
- Ideen her er at dersom en verdi blir hashet til noe som allerede er tatt
- Så flytter vi det til den neste ledige plassen

Linear probing - Prosedyre



Seperate Chaining VS Linear Probing

- Seperate Chaining er open hashing
- Linear probing er lukket hashing

Kan separate chaining implementeres med et BST for å få $\log(n)$ kjøretid på operasjonene sine?

19 responses

hvordan er load factor relatert til størrelsen på arrayen

vet ikke

Ja

heap?

jeg tror det

Ja

Ja. Java bruker vel rød svarte trær

ja

Vet ikke :P

Kan separate chaining implementeres med et BST for å få $\log(n)$ kjøretid på operasjonene sine?

19 responses

ja, fordi at det bare ligger en liste bak, det er ikke noe fancy

Vet ikke

På alle operasjoner?

ja

Ja :)

ja

vet ikke

ja, om dataen er lagret som noe som kan telles (enten lengden på str eller tallet på en int/double), så kan man få $\log(n)$ kjøretid fordi den halvererer lista for hvert steg

blir det ikke $O(n \cdot \log(n))$ fordi den må først gå til indeksen og så gå gjennom lista i den indeksen?

Kan separate chaining implementeres med et BST for å få $\log(n)$ kjøretid på operasjonene sine?

19 responses

nei -stellka

Rehashing

- Oppstår når datastrukturen vår er full
- Da må vi lage en ny datastruktur med dobbel lengde
- Og overføre elementene over

Gruppeoppgaver

Linear probing

2 poeng

Vi starter med et tomt array på størrelse 10.

0	1	2	3	4	5	6	7	8	9

Hashfunksjonen du skal bruke er $h(k, N) = k \bmod N$, som for dette eksempelet blir det samme som $h(k) = k \bmod 10$. Altså hasher et tall til sitt siste siffer.

Bruk *linear probing* til å sette inn disse tallene i den gitte rekkefølgen:

93, 48, 74, 99, 29, 13, 45

Fyll ut tabellen slik den ser ut etter alle tallene er satt inn med linear probing.

Eksamen 2020

Lukket Hashing(Eksamen 2019 modifisert)

Gitt en hashtabell av lengde 5 som skal lagre heltall.

Vi bruker lukket hashing med linear probing og hashfunksjonen $h(k) = k \bmod 5$. (k modulo 5)

Vi legger følgende elementer(nøkler) inn i tabellen i denne rekkefølgen:

17, 98, 59, 32, 40.

På hvilken indeks havner siste element(40)?

Hvilken index havner 40 på dersom vi bruker separate chaining?

Spørsmål

0 questions

0 upvotes

