



# DM503

Forelæsning 2



# Indhold



# Indhold

- Introduktion til ADT'er (Abstrakte DataTyper)





# Indhold

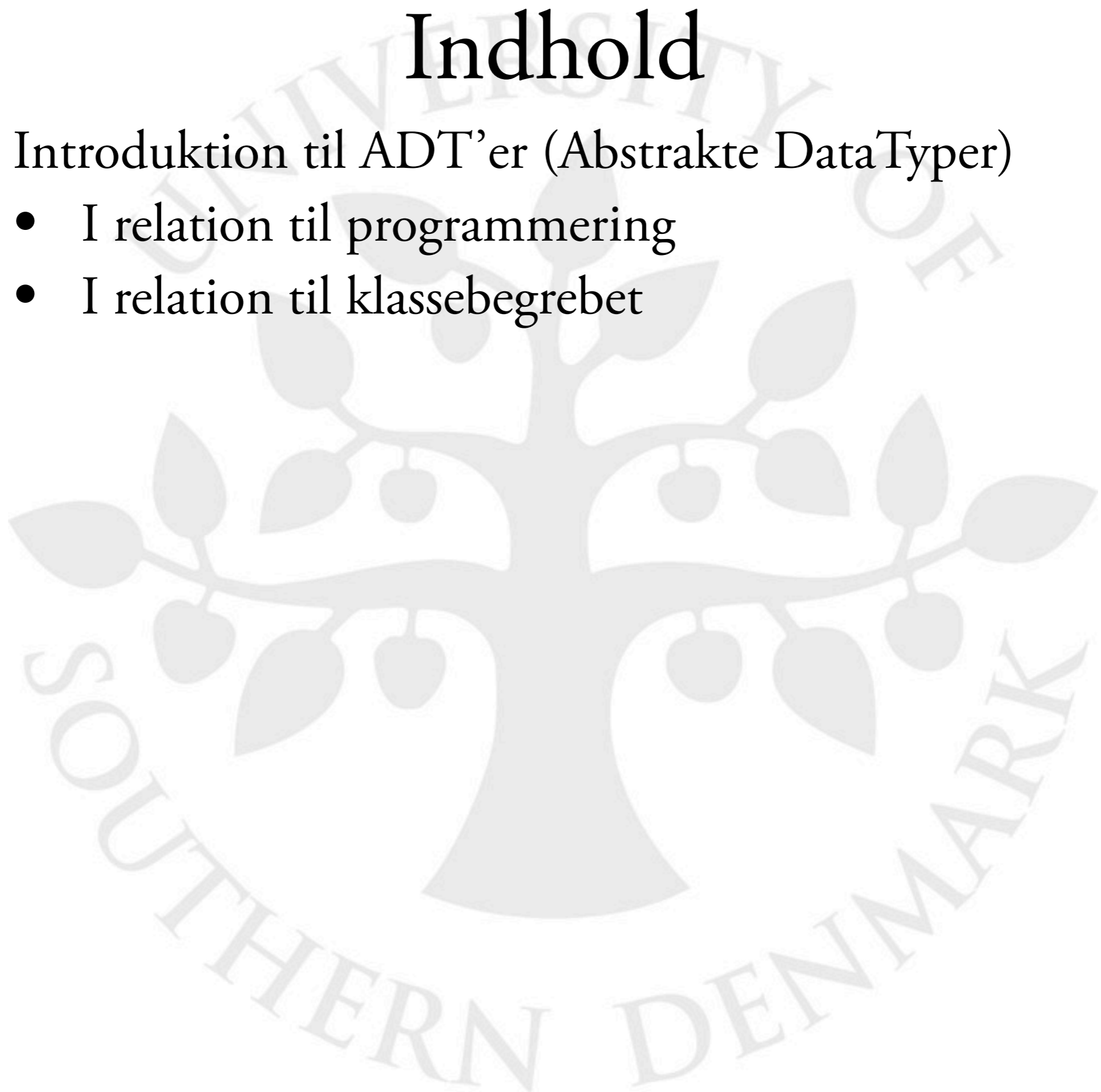
- Introduktion til ADT'er (Abstrakte DataTyper)
  - I relation til programmering





# Indhold

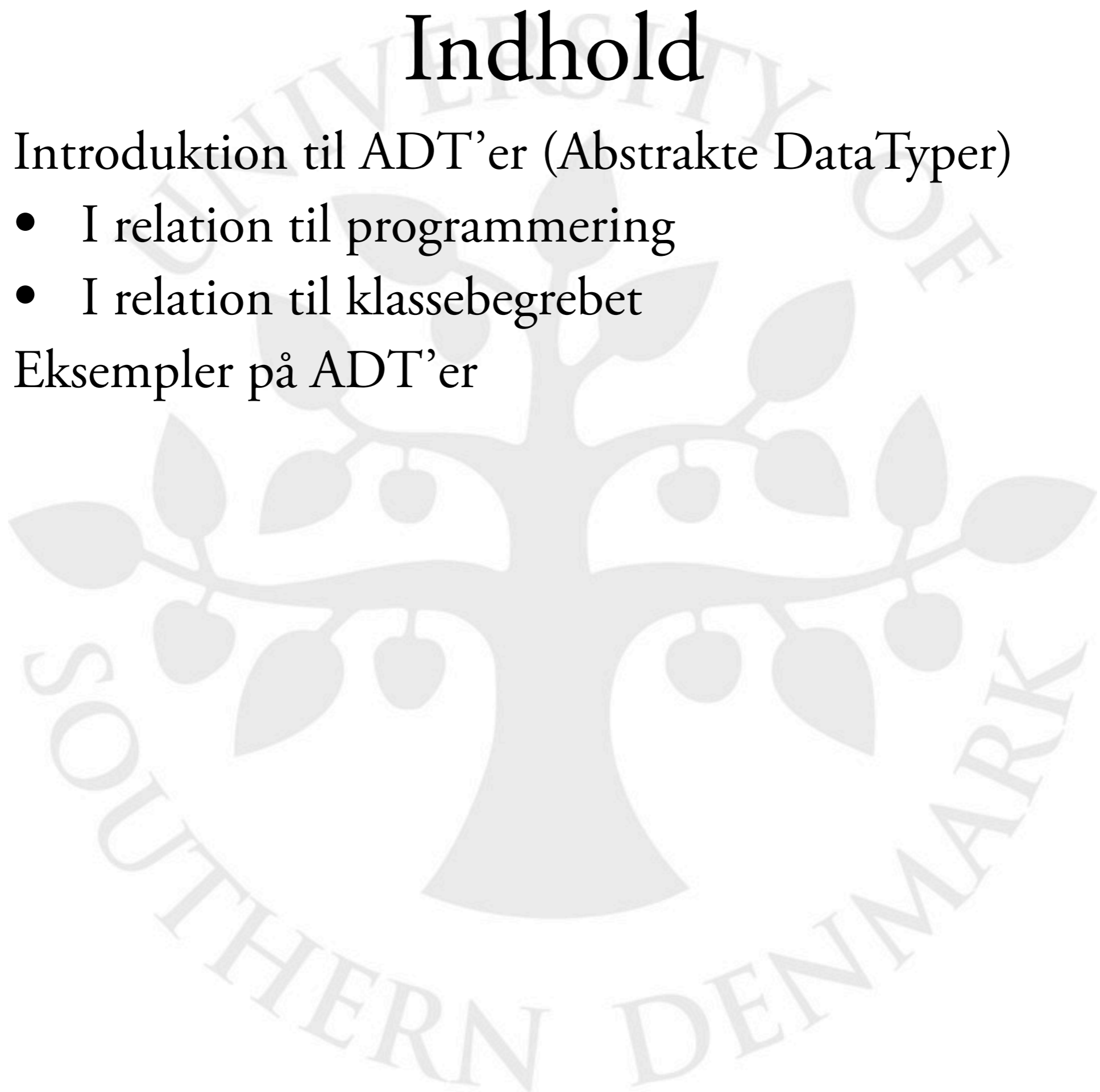
- Introduktion til ADT'er (Abstrakte DataTyper)
  - I relation til programmering
  - I relation til klassebegrebet





# Indhold

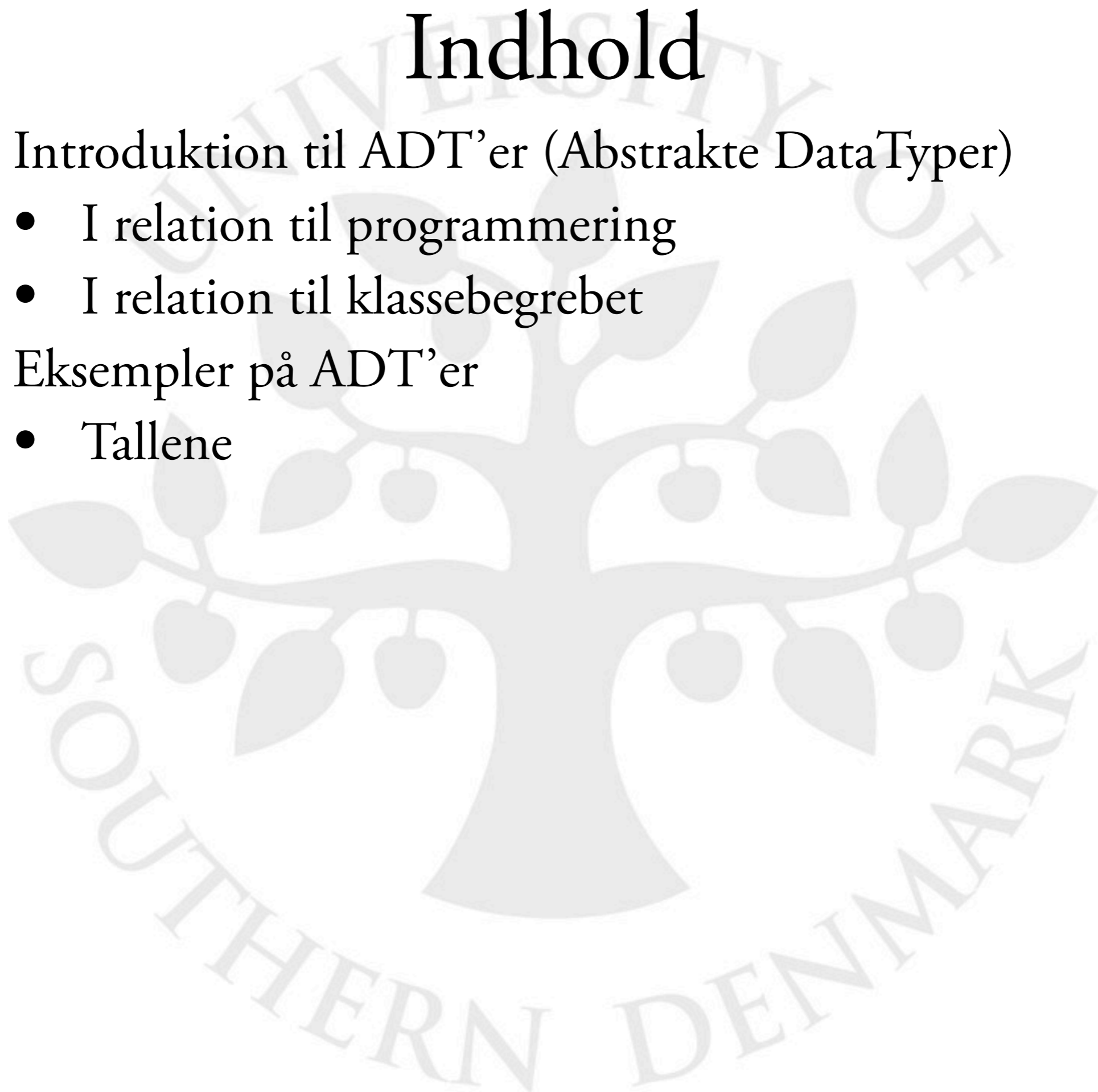
- Introduktion til ADT'er (Abstrakte DataTyper)
  - I relation til programmering
  - I relation til klassebegrebet
- Eksempler på ADT'er





# Indhold

- Introduktion til ADT'er (Abstrakte DataTyper)
  - I relation til programmering
  - I relation til klassebegrebet
- Eksempler på ADT'er
  - Tallene



# Indhold

- Introduktion til ADT'er (Abstrakte DataTyper)
  - I relation til programmering
  - I relation til klassebegrebet
- Eksempler på ADT'er
  - Tallene
  - Følger





# Indhold

- Introduktion til ADT'er (Abstrakte DataTyper)
  - I relation til programmering
  - I relation til klassebegrebet
- Eksempler på ADT'er
  - Tallene
  - Følger
    - Stakke



# Indhold

- Introduktion til ADT'er (Abstrakte DataTyper)
  - I relation til programmering
  - I relation til klassebegrebet
- Eksempler på ADT'er
  - Tallene
  - Følger
    - Stakke
    - Køer

# Abstrakte DataTyper



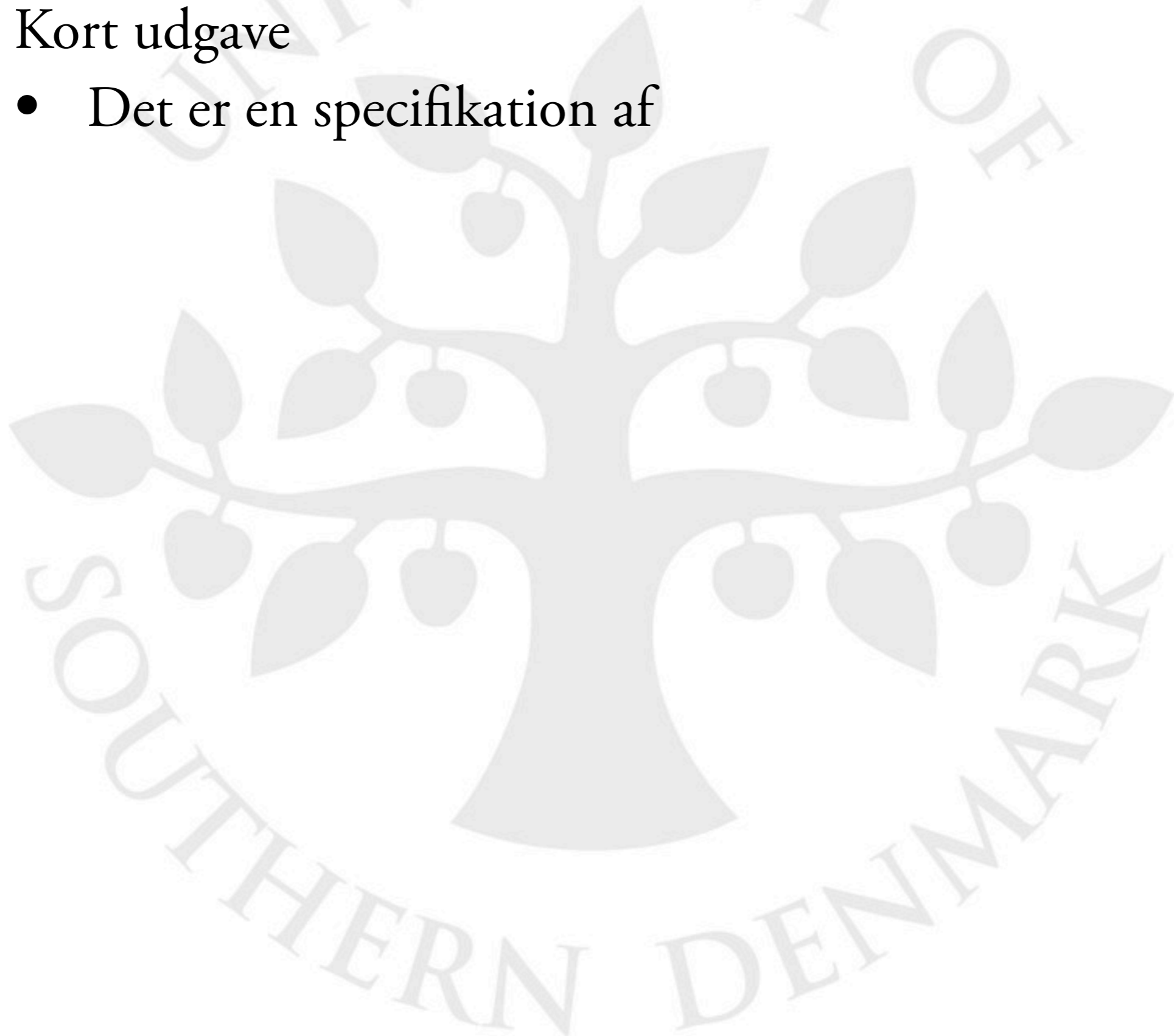
# Abstrakte DataTyper

- Kort udgave



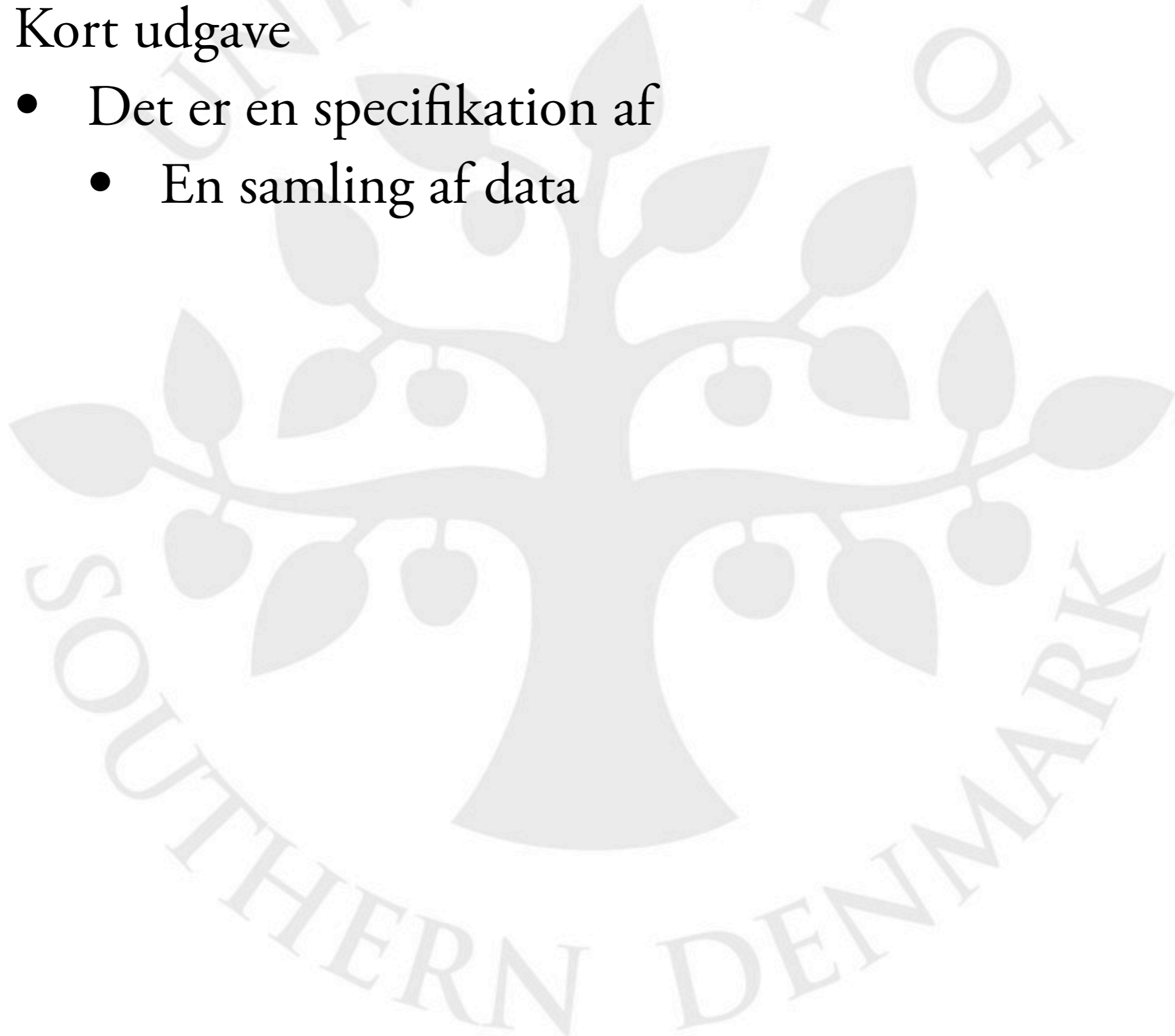
# Abstrakte DataTyper

- Kort udgave
  - Det er en specifikation af



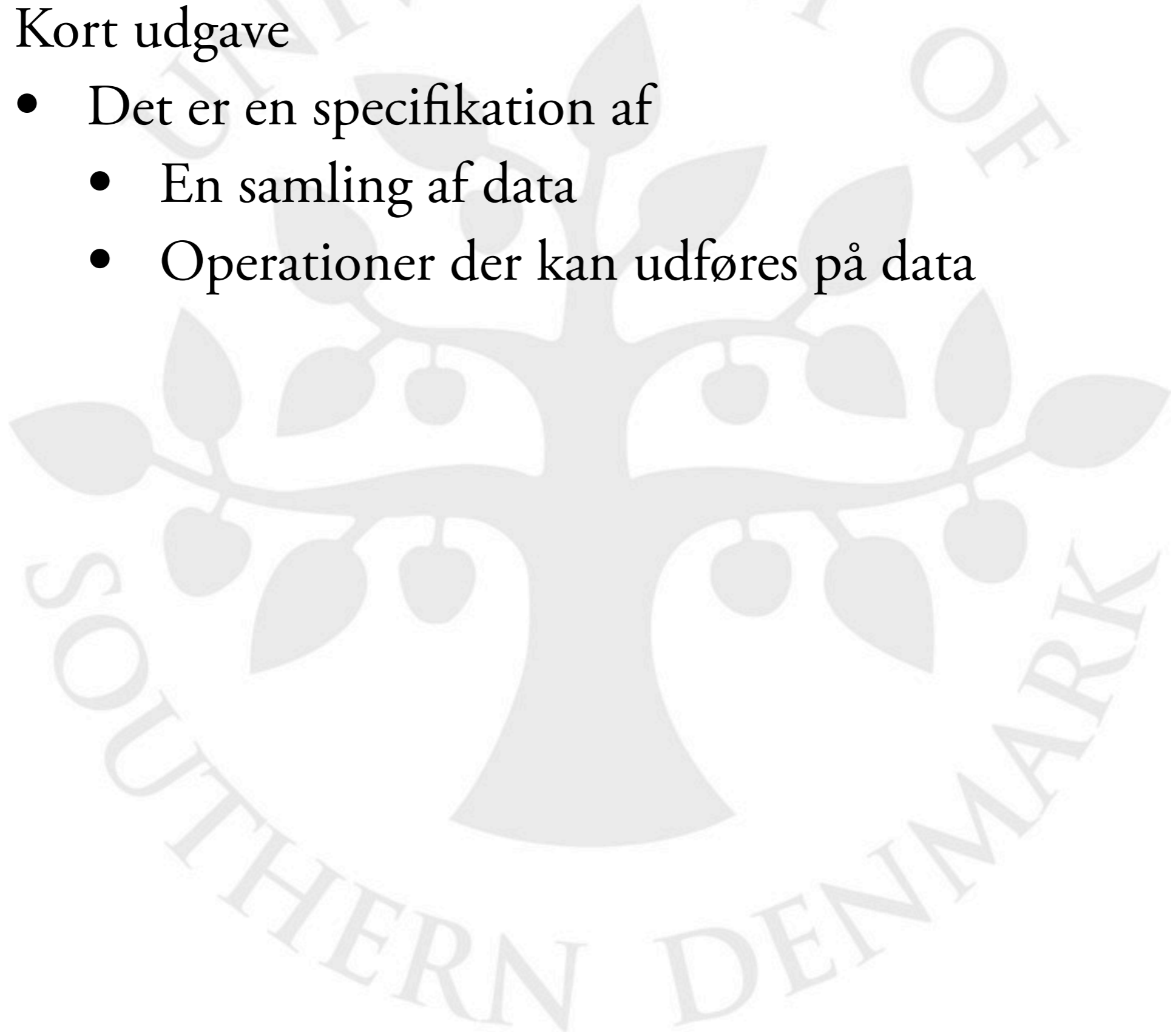
# Abstrakte DataTyper

- Kort udgave
  - Det er en specifikation af
    - En samling af data



# Abstrakte DataTyper

- Kort udgave
  - Det er en specifikation af
    - En samling af data
    - Operationer der kan udføres på data



# Abstrakte DataTyper

- Kort udgave
  - Det er en specifikation af
    - En samling af data
    - Operationer der kan udføres på data
      - Indeholder altid mindst én constructor





# Abstrakte DataTyper

- Kort udgave
  - Det er en specifikation af
    - En samling af data
    - Operationer der kan udføres på data
      - Indeholder altid mindst én constructor
      - Manipulationer



# Abstrakte DataTyper

- Kort udgave
  - Det er en specifikation af
    - En samling af data
    - Operationer der kan udføres på data
      - Indeholder altid mindst én constructor
      - Manipulationer
      - Aflæsninger



# Abstrakte DataTyper

- Kort udgave
  - Det er en specifikation af
    - En samling af data
    - Operationer der kan udføres på data
      - Indeholder altid mindst én constructor
      - Manipulationer
      - Aflæsninger
      - Beregninger

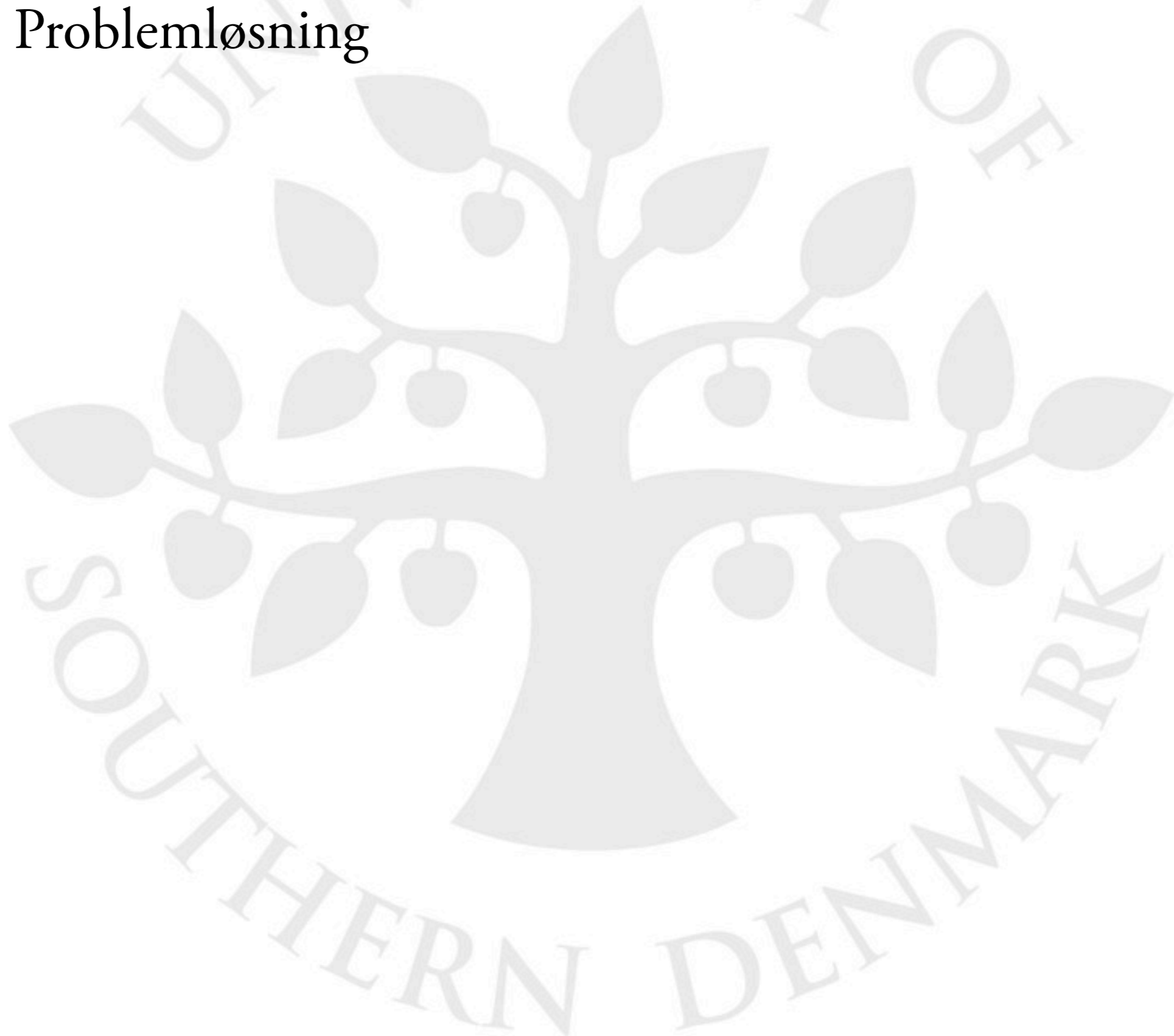


# Abstrakte DataTyper



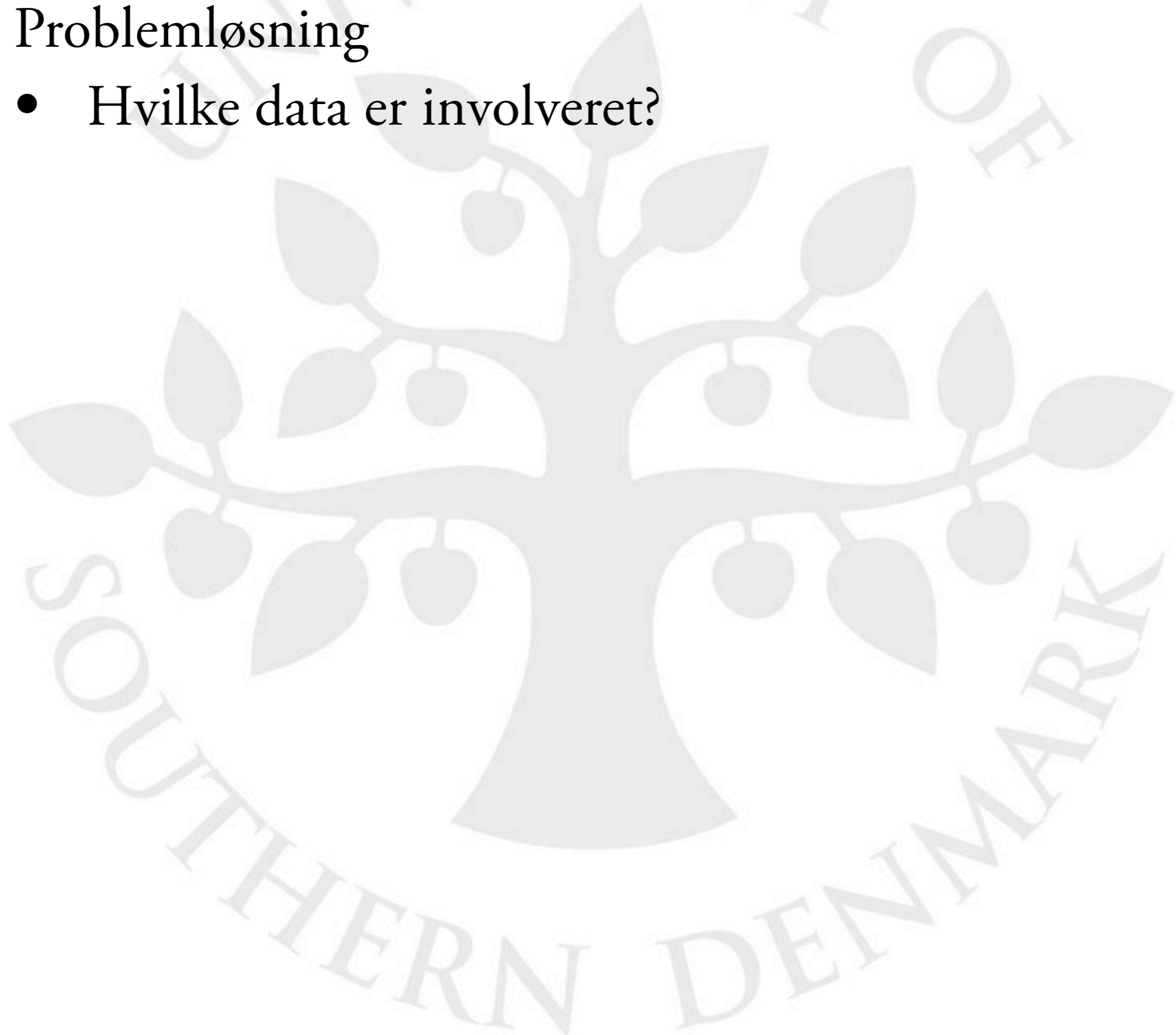
# Abstrakte DataTyper

- Problemløsning



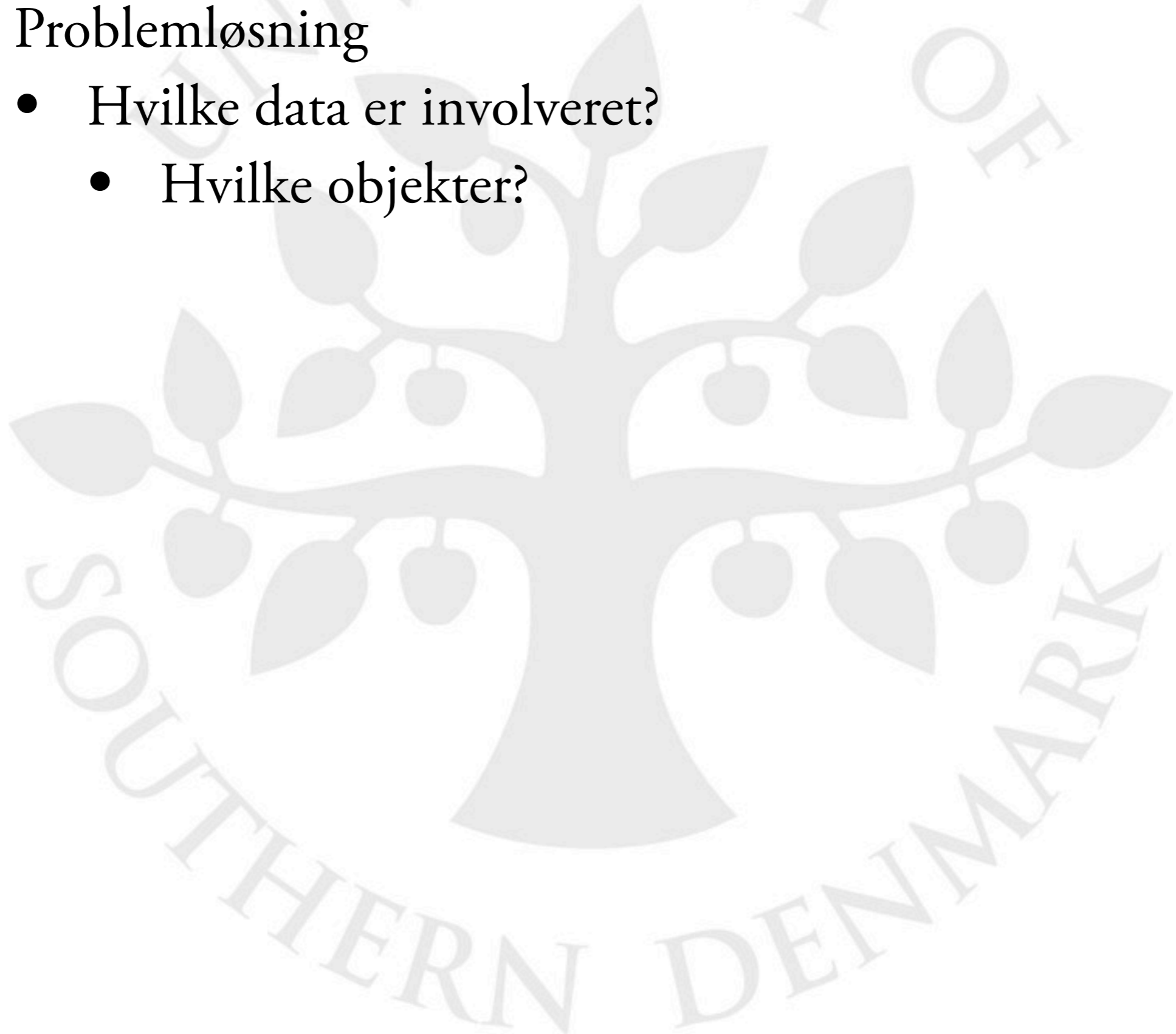
# Abstrakte DataTyper

- Problemløsning
  - Hvilke data er involveret?



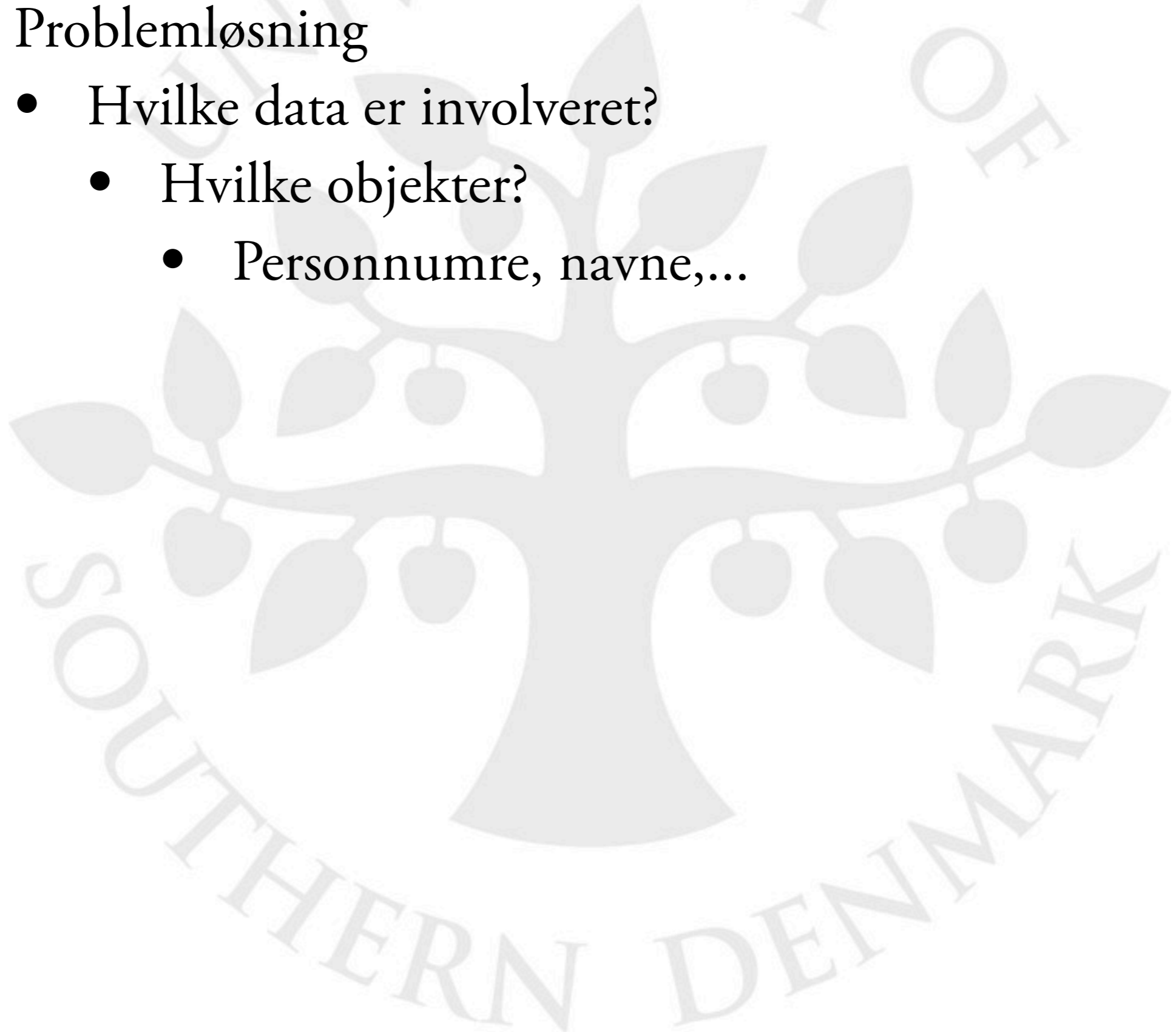
# Abstrakte DataTyper

- Problemløsning
  - Hvilke data er involveret?
    - Hvilke objekter?



# Abstrakte DataTyper

- Problemløsning
  - Hvilke data er involveret?
    - Hvilke objekter?
      - Personnumre, navne,...





# Abstrakte DataTyper

- Problemløsning
  - Hvilke data er involveret?
    - Hvilke objekter?
      - Personnumre, navne,...
  - Hvad vil vi gerne kunne gøre ved data?



# Abstrakte DataTyper

- Problemløsning
  - Hvilke data er involveret?
    - Hvilke objekter?
      - Personnumre, navne,...
  - Hvad vil vi gerne kunne gøre ved data?
    - Hvilke operationer?



# Abstrakte DataTyper

- Problemløsning
  - Hvilke data er involveret?
    - Hvilke objekter?
      - Personnumre, navne,...
  - Hvad vil vi gerne kunne gøre ved data?
    - Hvilke operationer?
      - Finde de personnumre der er født mellem dato X og dato Y

# Abstrakte DataTyper

- Problemløsning
  - Hvilke data er involveret?
    - Hvilke objekter?
      - Personnumre, navne,...
  - Hvad vil vi gerne kunne gøre ved data?
    - Hvilke operationer?
      - Finde de personnumre der er født mellem dato X og dato Y
      - Udskrive navnene i sorteret rækkefølge efter efternavn

# Abstrakte DataTyper

- Problemløsning
  - Hvilke data er involveret?
    - Hvilke objekter?
      - Personnumre, navne,...
  - Hvad vil vi gerne kunne gøre ved data?
    - Hvilke operationer?
      - Finde de personnumre der er født mellem dato X og dato Y
      - Udskrive navnene i sorteret rækkefølge efter efternavn
      - ...

# Abstrakte DataTyper

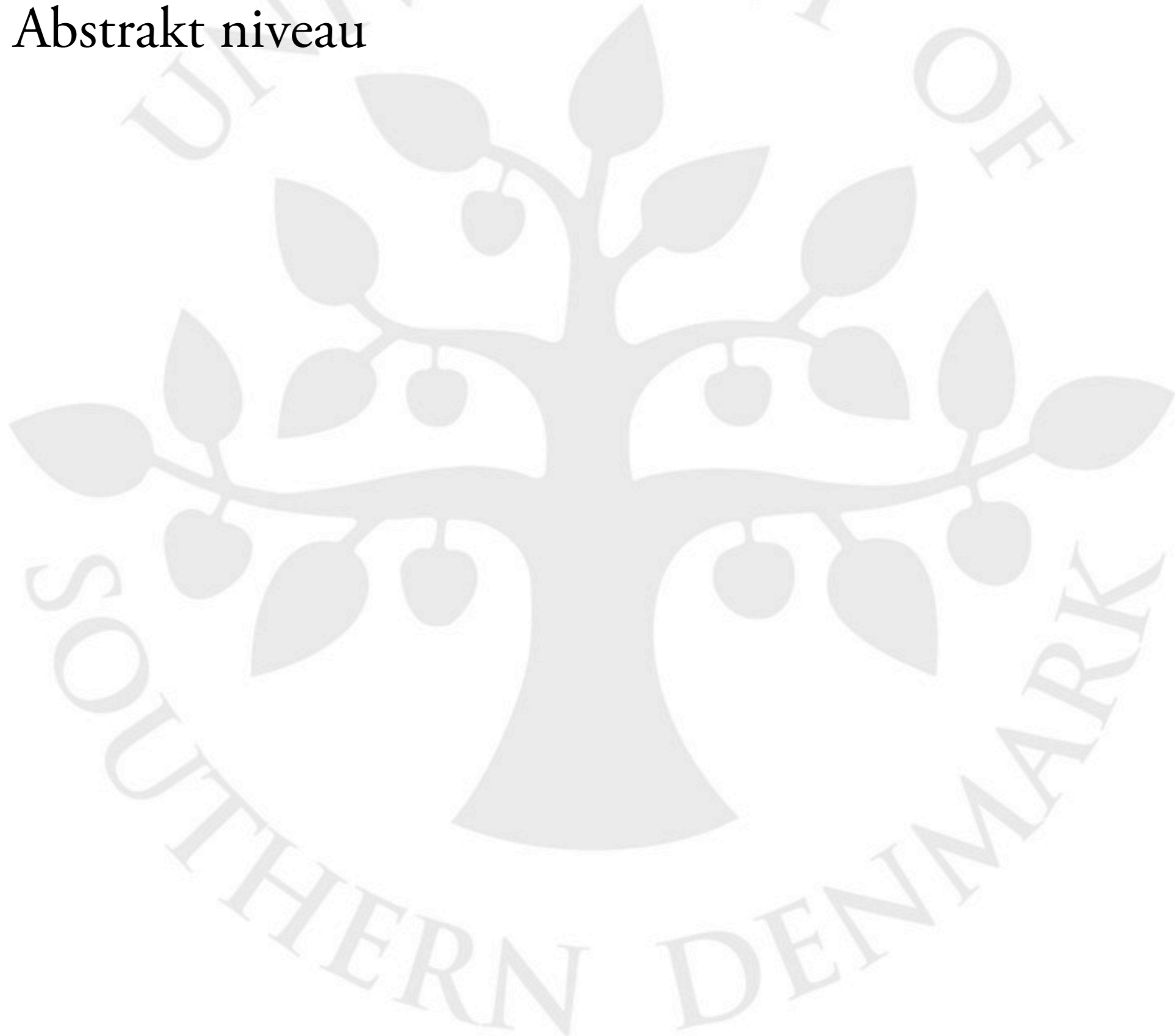
- Problemløsning
  - Hvilke data er involveret?
    - Hvilke objekter?
      - Personnumre, navne,...
  - Hvad vil vi gerne kunne gøre ved data?
    - Hvilke operationer?
      - Finde de personnumre der er født mellem dato X og dato Y
      - Udskrive navnene i sorteret rækkefølge efter efternavn
      - ...
- Analyse af problemet, giver de værktøjer vi har brug for at løse det

# Abstrakte DataTyper



# Abstrakte DataTyper

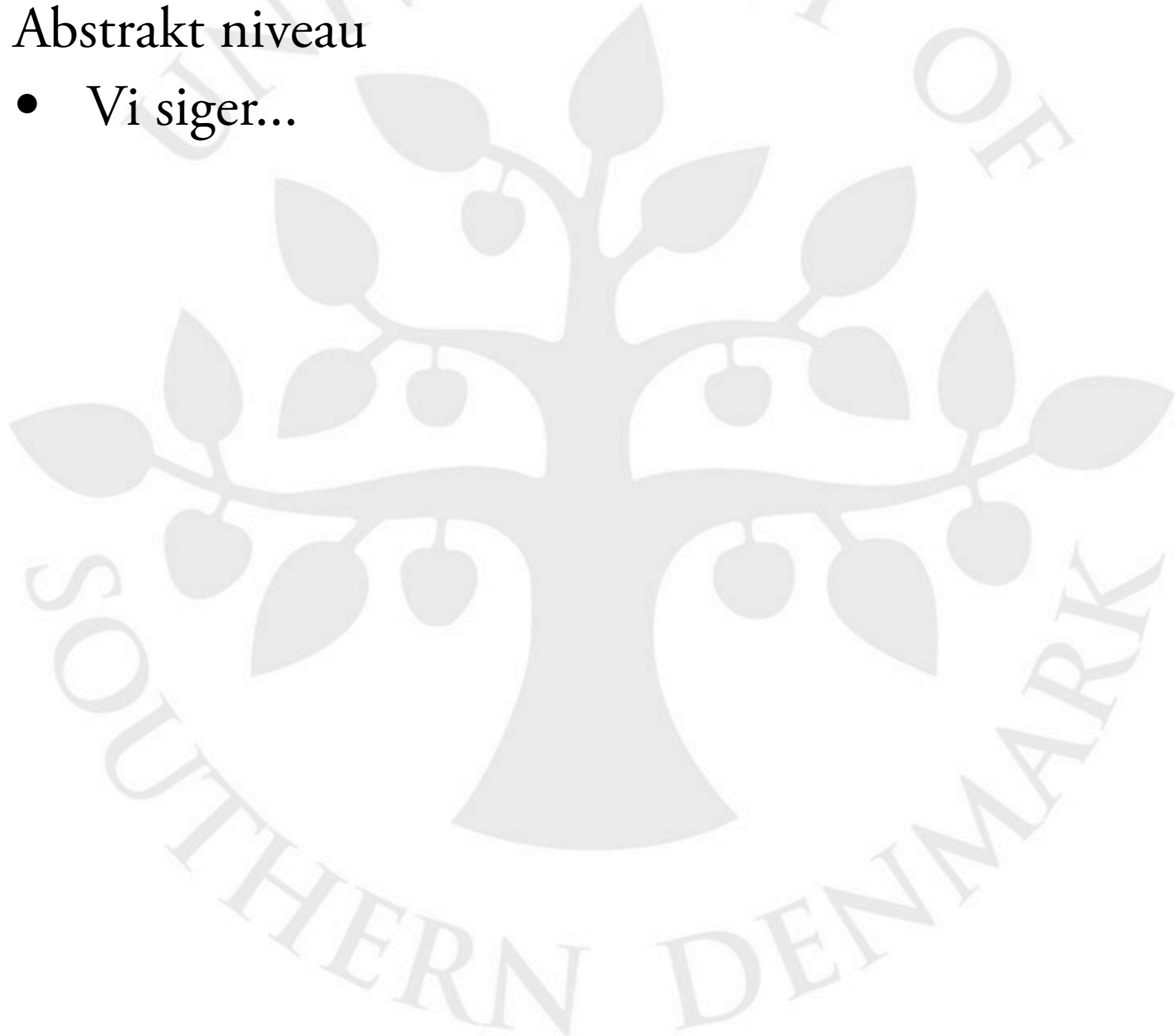
- Abstrakt niveau





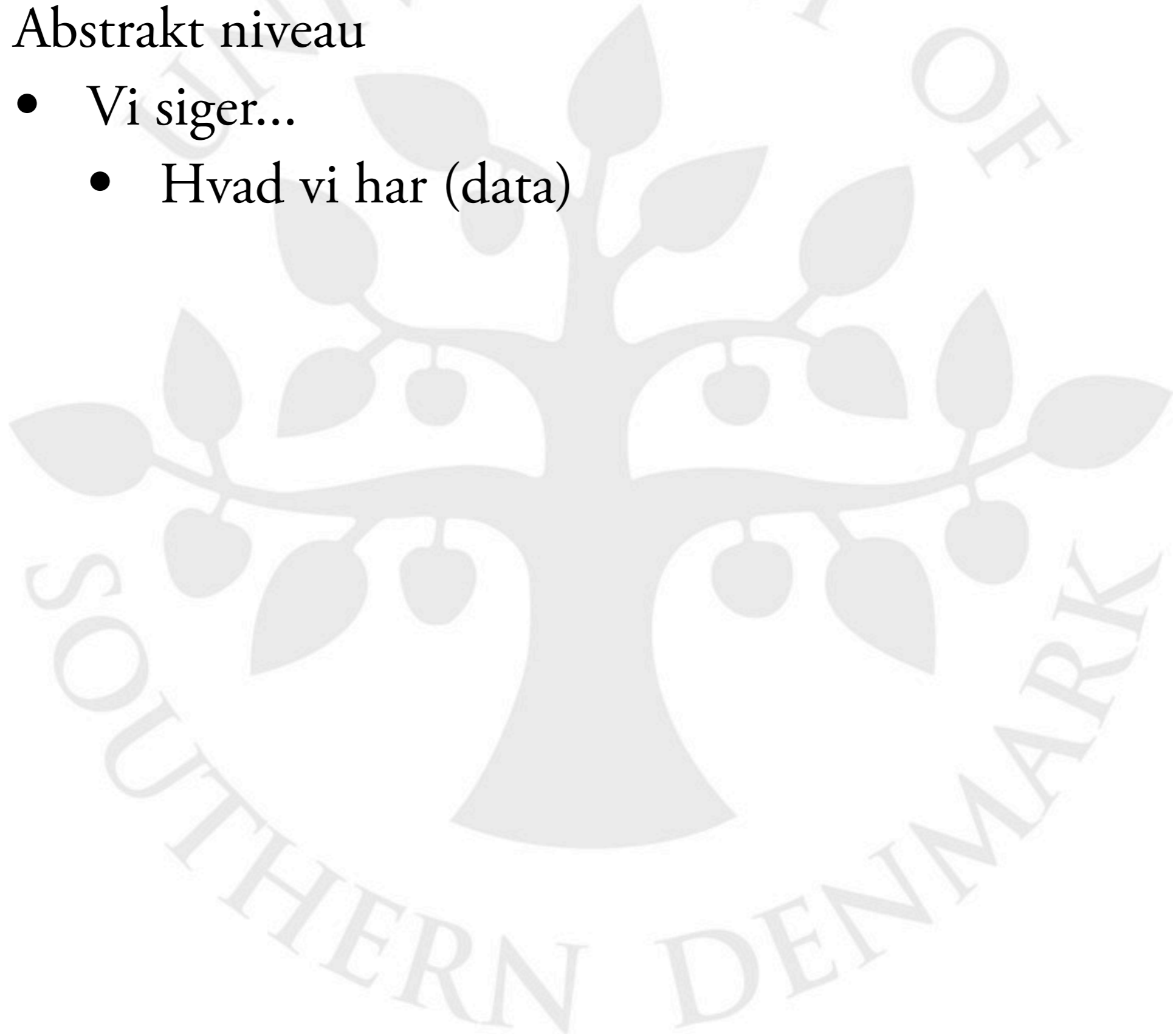
# Abstrakte DataTyper

- Abstrakt niveau
  - Vi siger...



# Abstrakte DataTyper

- Abstrakt niveau
  - Vi siger...
  - Hvad vi har (data)



# Abstrakte DataTyper

- Abstrakt niveau
  - Vi siger...
    - Hvad vi har (data)
    - Hvad vi gerne vil kunne gøre (operationer)



# Abstrakte DataTyper

- Abstrakt niveau
  - Vi siger...
    - Hvad vi har (data)
    - Hvad vi gerne vil kunne gøre (operationer)
    - Hvad resultatet af disse operationer skal være



# Abstrakte DataTyper

- Abstrakt niveau
  - Vi siger...
    - Hvad vi har (data)
    - Hvad vi gerne vil kunne gøre (operationer)
    - Hvad resultatet af disse operationer skal være
  - Vi siger ikke

# Abstrakte DataTyper

- Abstrakt niveau
  - Vi siger...
    - Hvad vi har (data)
    - Hvad vi gerne vil kunne gøre (operationer)
    - Hvad resultatet af disse operationer skal være
  - Vi siger ikke
    - Hvor og hvordan data skal ligge



# Abstrakte DataTyper

- Abstrakt niveau
  - Vi siger...
    - Hvad vi har (data)
    - Hvad vi gerne vil kunne gøre (operationer)
    - Hvad resultatet af disse operationer skal være
  - Vi siger ikke
    - Hvor og hvordan data skal ligge
    - Hvordan operationerne skal laves

# Abstrakte DataTyper

- Abstrakt niveau
  - Vi siger...
    - Hvad vi har (data)
    - Hvad vi gerne vil kunne gøre (operationer)
    - Hvad resultatet af disse operationer skal være
  - Vi siger ikke
    - Hvor og hvordan data skal ligge
    - Hvordan operationerne skal laves
- Uafhængigt af implementeringen!



# Abstrakte DataTyper

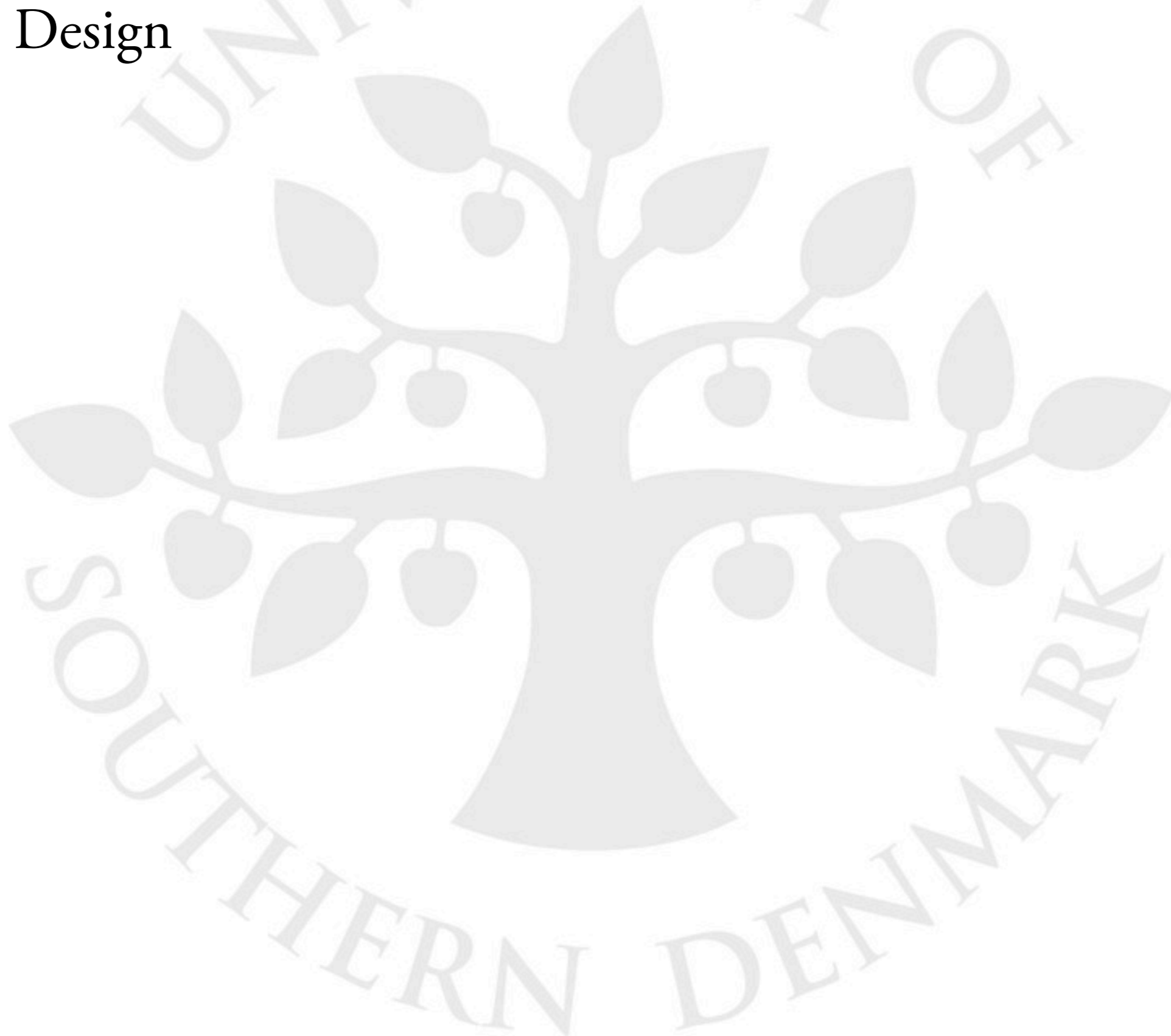
- Abstrakt niveau
  - Vi siger...
    - Hvad vi har (data)
    - Hvad vi gerne vil kunne gøre (operationer)
    - Hvad resultatet af disse operationer skal være
  - Vi siger ikke
    - Hvor og hvordan data skal ligge
    - Hvordan operationerne skal laves
- Uafhængigt af implementeringen!
  - Flere skridt fra at skrive en linje kode

# Abstrakte DataTyper



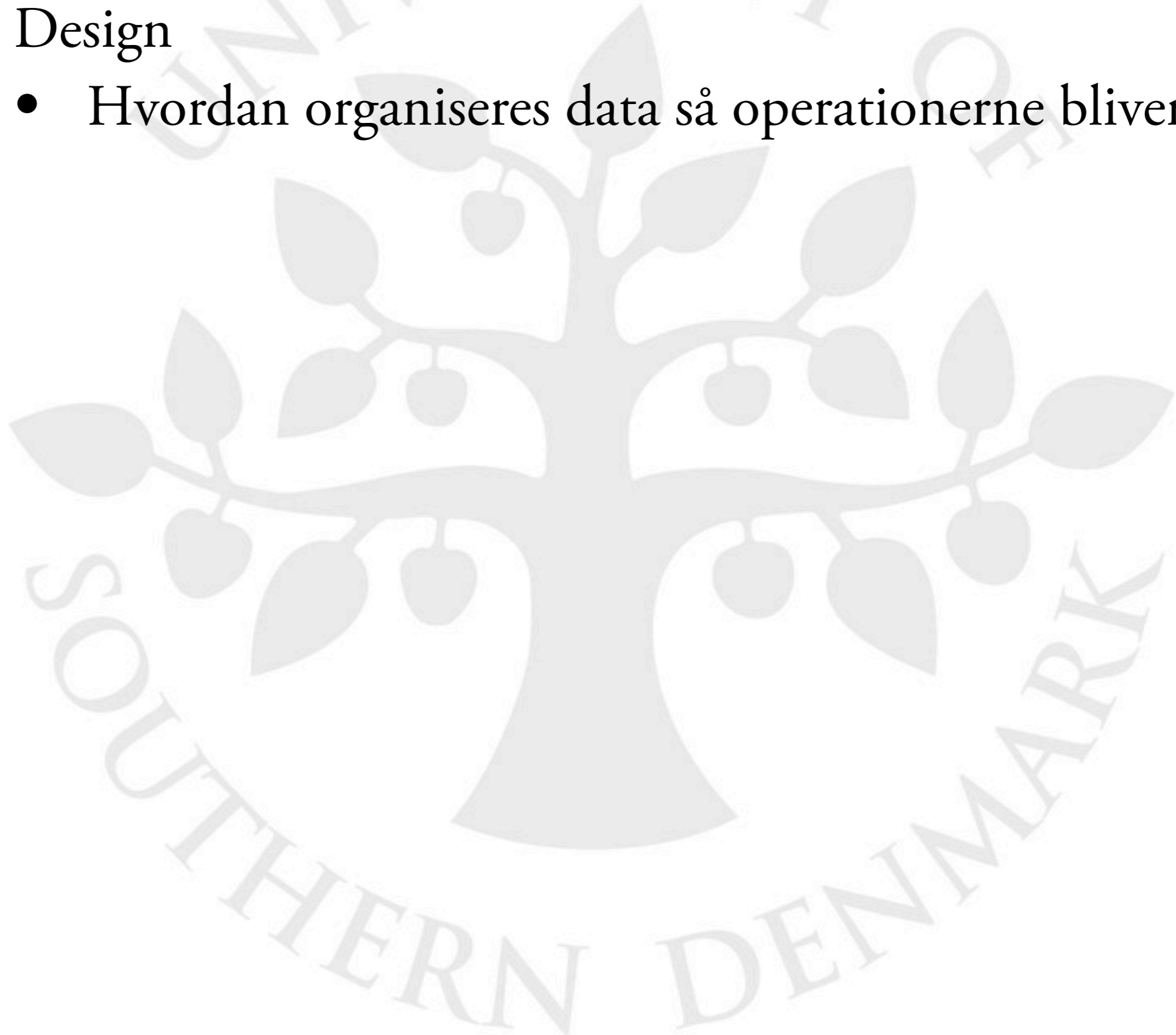
# Abstrakte DataTyper

- Design



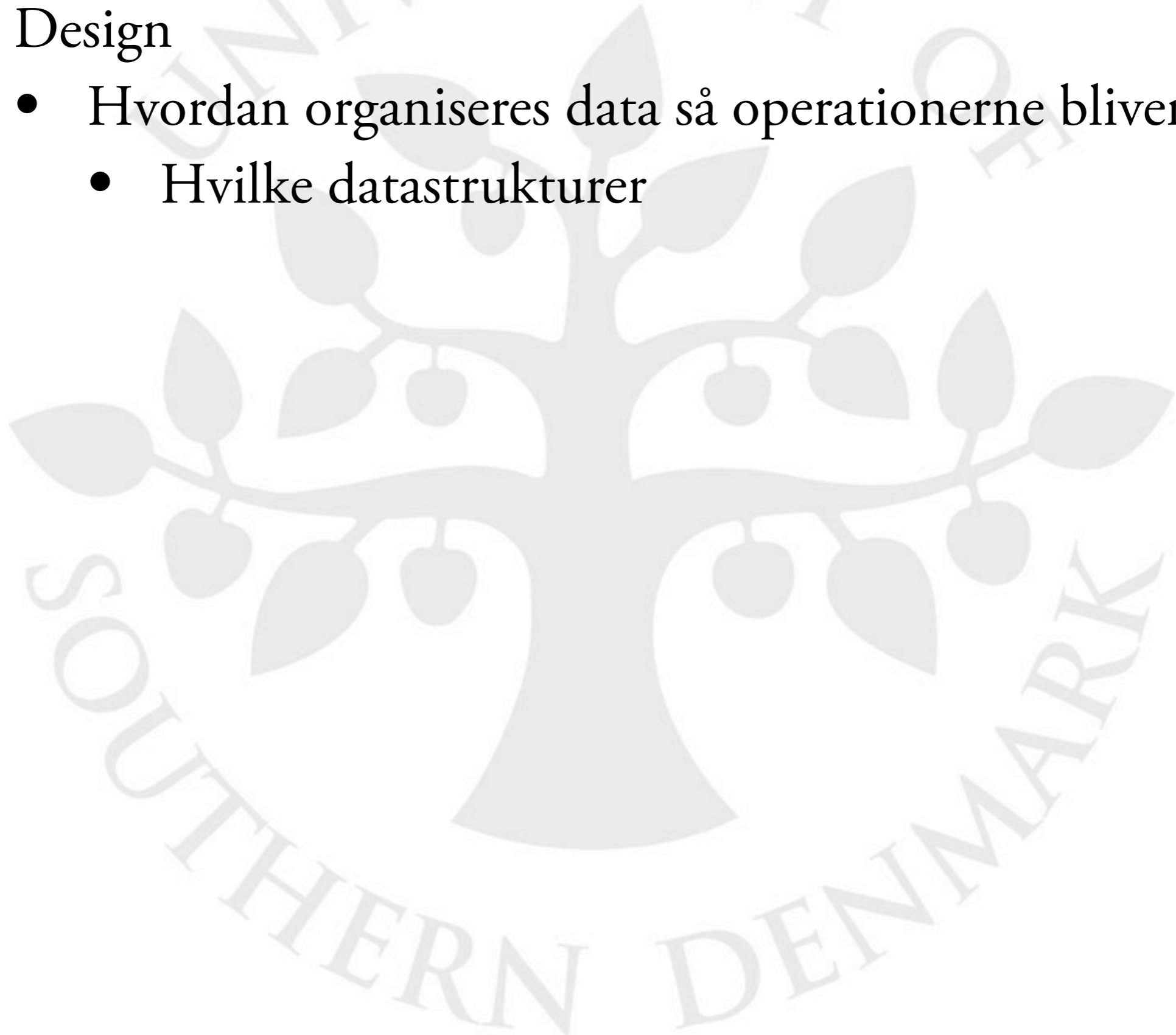
# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige



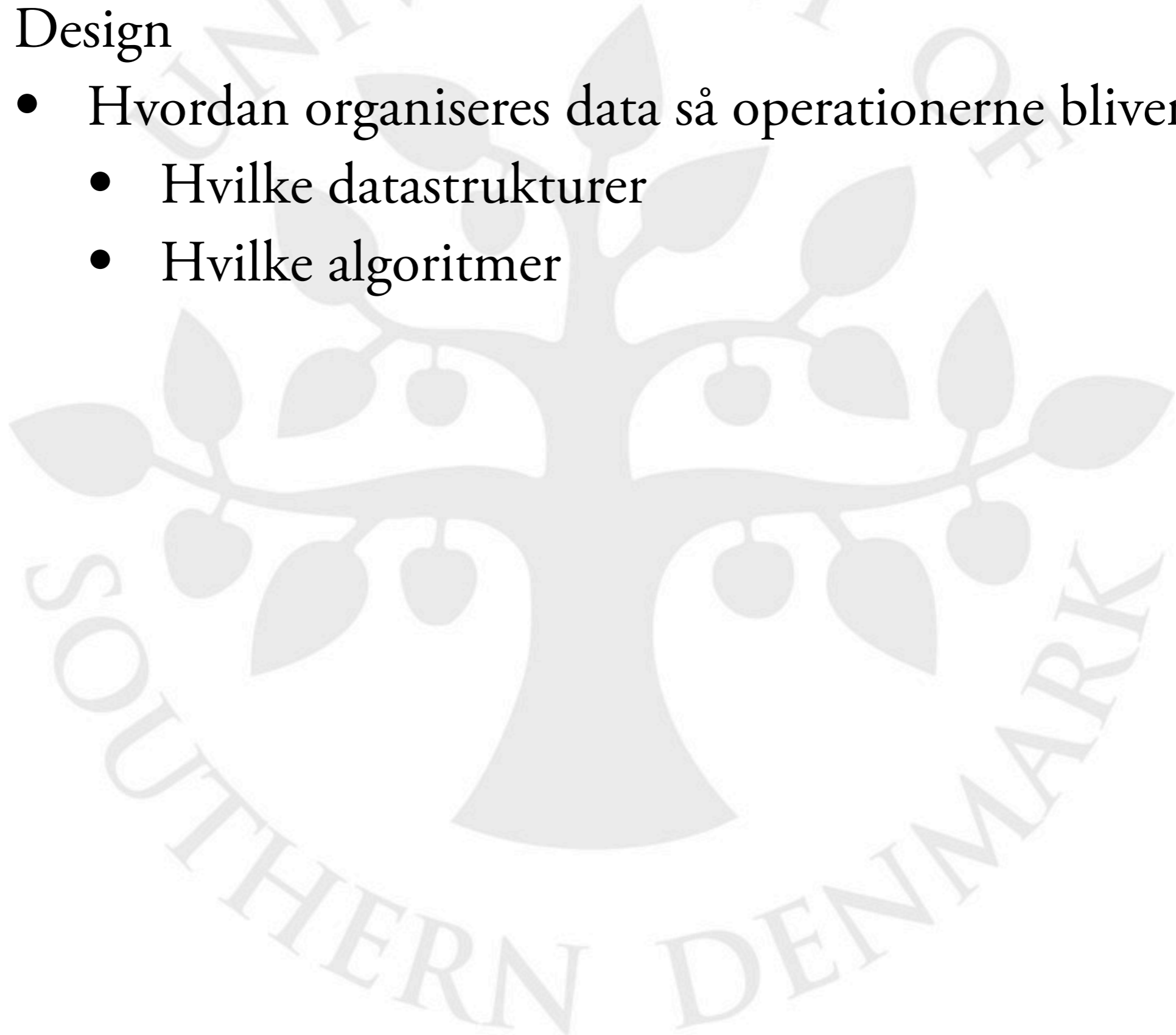
# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer



# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer
    - Hvilke algoritmer



# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer
    - Hvilke algoritmer
  - Stadig abstrakt



# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer
    - Hvilke algoritmer
  - Stadig abstrakt
    - Uafhængigt af programmeringssproget





# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer
    - Hvilke algoritmer
  - Stadig abstrakt
    - Uafhængigt af programmeringssproget
  - Bekymringer



# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer
    - Hvilke algoritmer
  - Stadig abstrakt
    - Uafhængigt af programmeringssproget
  - Bekymringer
    - Effektivitet!

# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer
    - Hvilke algoritmer
  - Stadig abstrakt
    - Uafhængigt af programmeringssproget
  - Bekymringer
    - Effektivitet!
    - Kompromiser

# Abstrakte DataTyper

- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer
    - Hvilke algoritmer
  - Stadig abstrakt
    - Uafhængigt af programmeringssproget
  - Bekymringer
    - Effektivitet!
    - Kompromiser
      - Ikke alle operationer kan eksistere på samme tid

# Abstrakte DataTyper

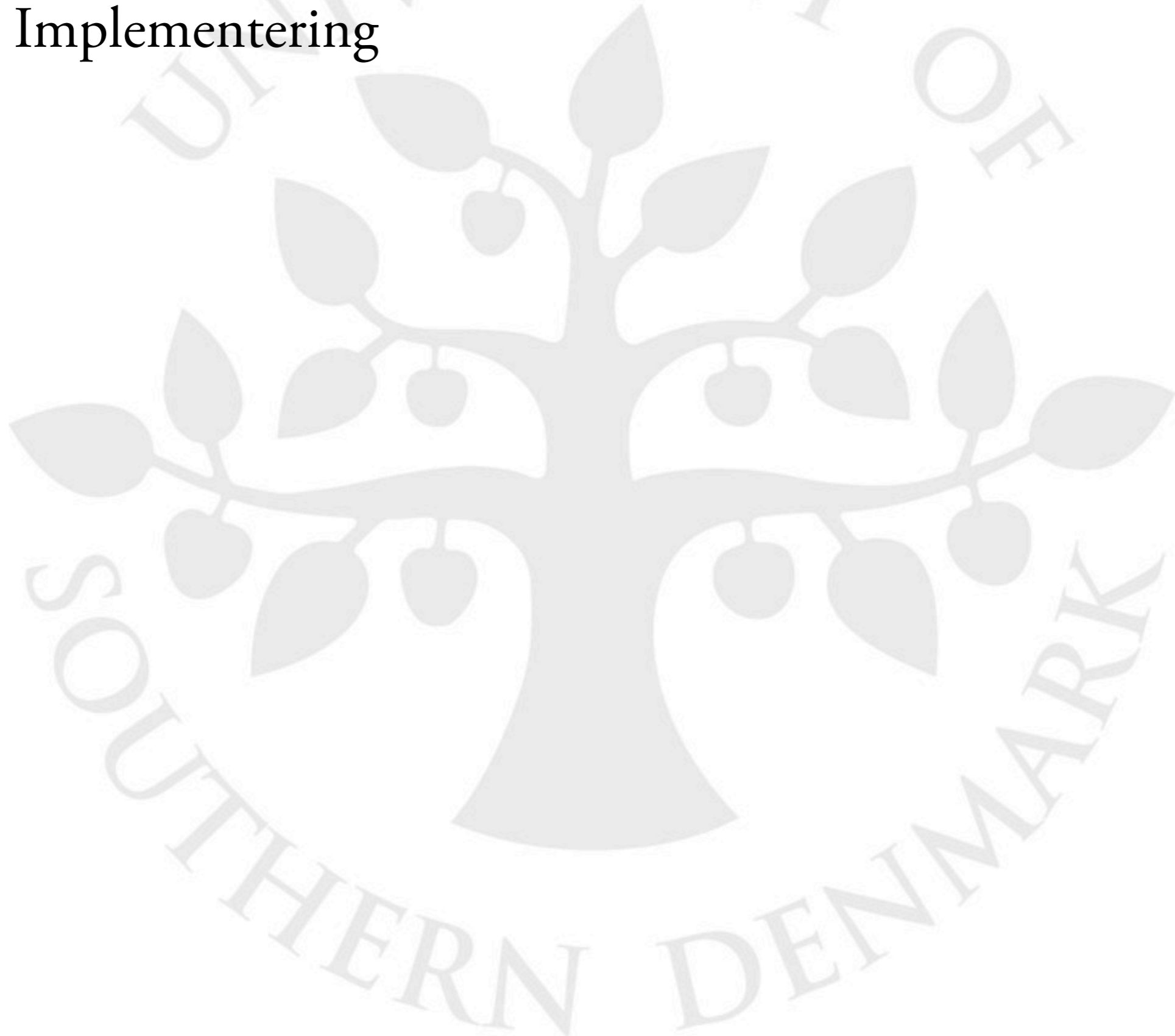
- Design
  - Hvordan organiseres data så operationerne bliver mulige
    - Hvilke datastrukturer
    - Hvilke algoritmer
  - Stadig abstrakt
    - Uafhængigt af programmeringssproget
- Bekymringer
  - Effektivitet!
  - Kompromiser
    - Ikke alle operationer kan eksistere på samme tid
    - Hvis “operation1” er effektiv kan “operation2” ikke også være effektiv

# Abstrakte DataTyper



# Abstrakte DataTyper

- Implementering



# Abstrakte DataTyper

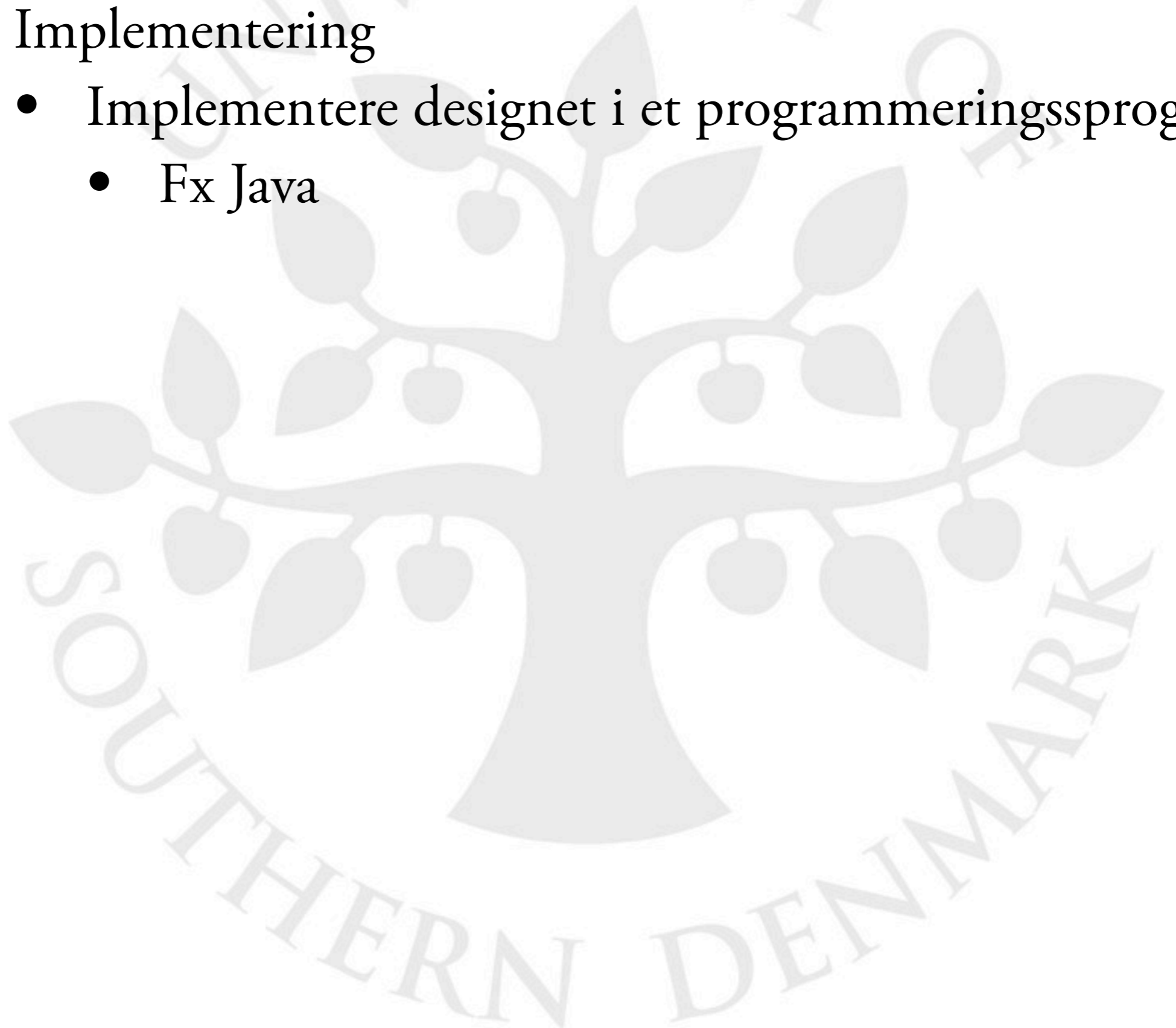
- Implementering
  - Implementere designet i et programmeringsprog





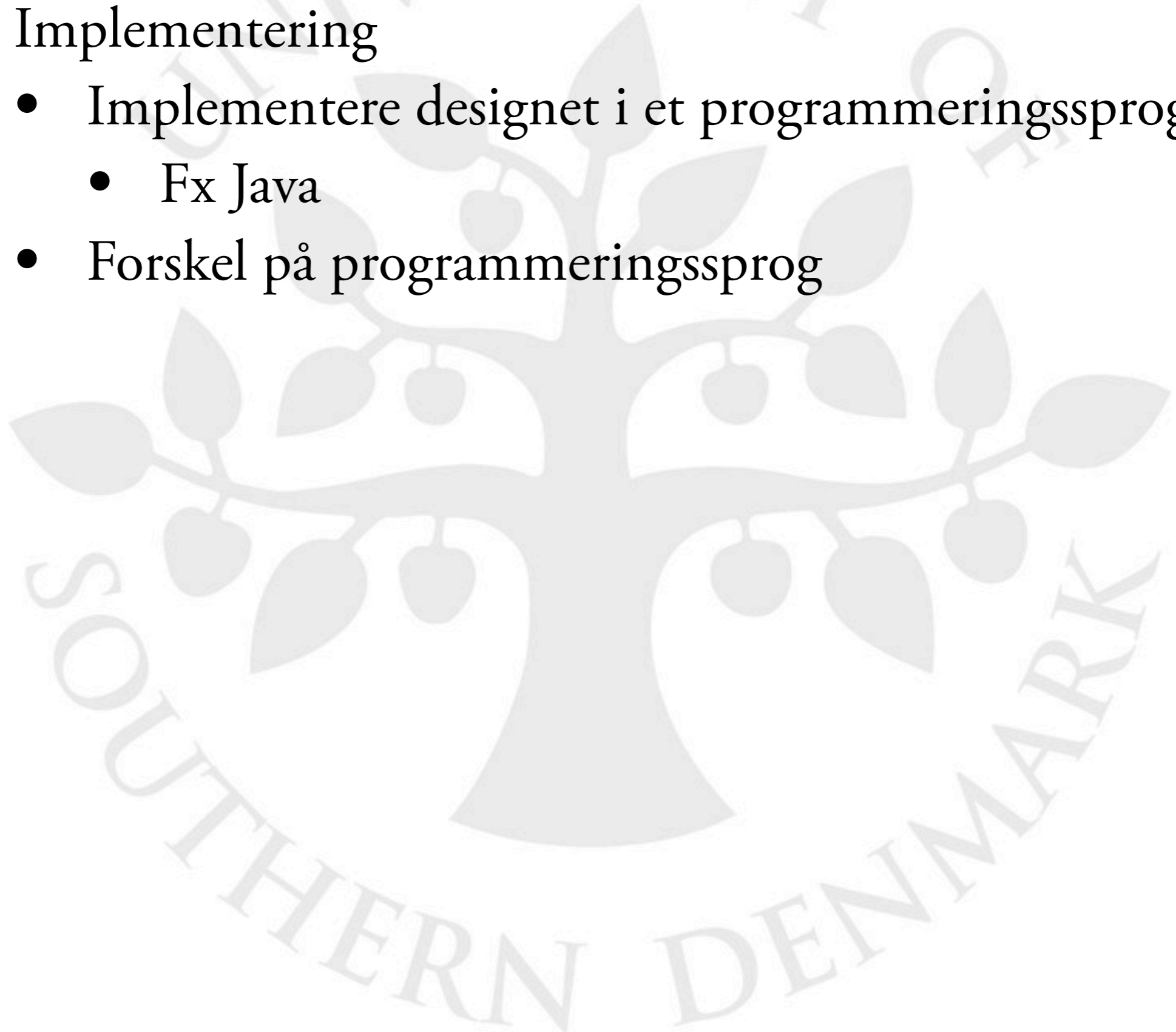
# Abstrakte DataTyper

- Implementering
  - Implementere designet i et programmeringsprog
    - Fx Java



# Abstrakte DataTyper

- Implementering
  - Implementere designet i et programmeringssprog
    - Fx Java
  - Forskel på programmeringssprog



# Abstrakte DataTyper

- Implementering
  - Implementere designet i et programmeringssprog
    - Fx Java
  - Forskel på programmeringssprog
    - Forskellige styrker og svagheder



# Abstrakte DataTyper

- Implementering
  - Implementere designet i et programmeringssprog
    - Fx Java
  - Forskel på programmeringssprog
    - Forskellige styrker og svagheder
    - “When all you have is a hammer, everything looks like a nail”



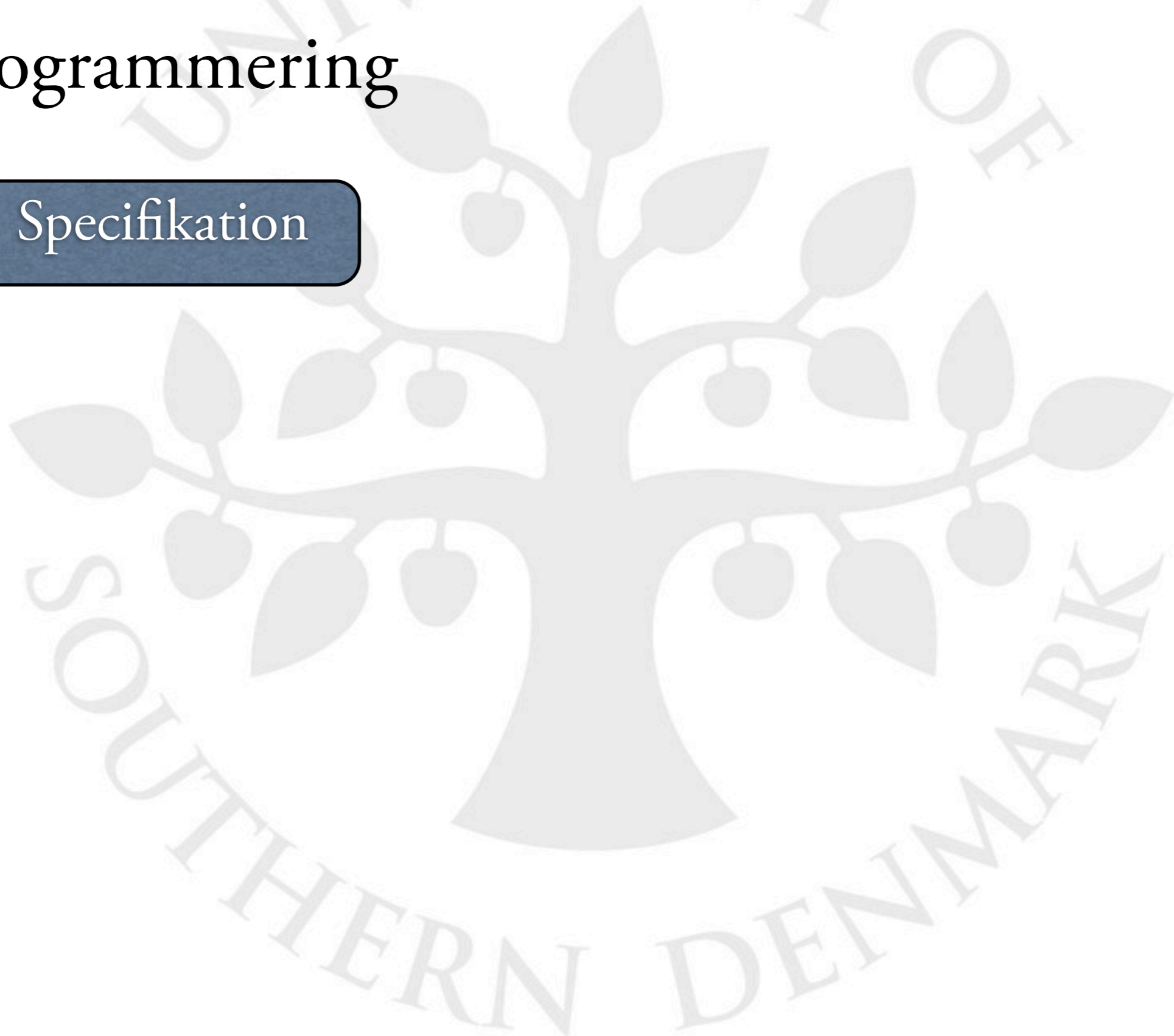
# Abstrakte DataTyper



# Abstrakte DataTyper

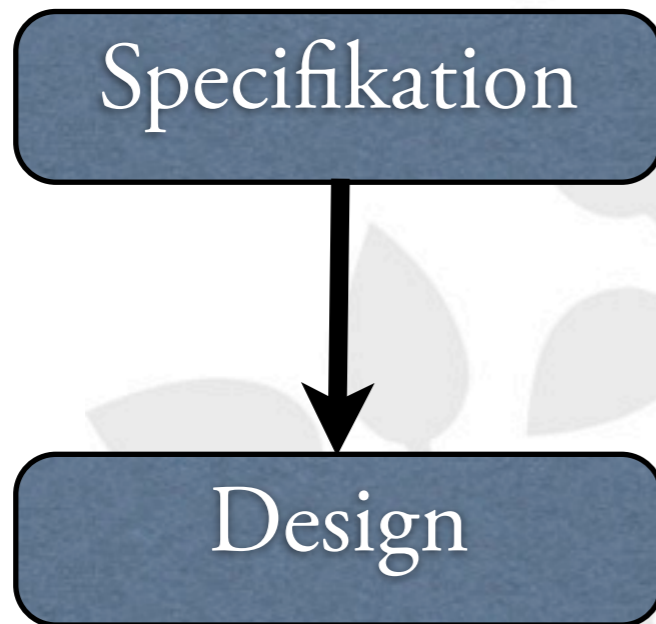
## Programmering

Specifikation



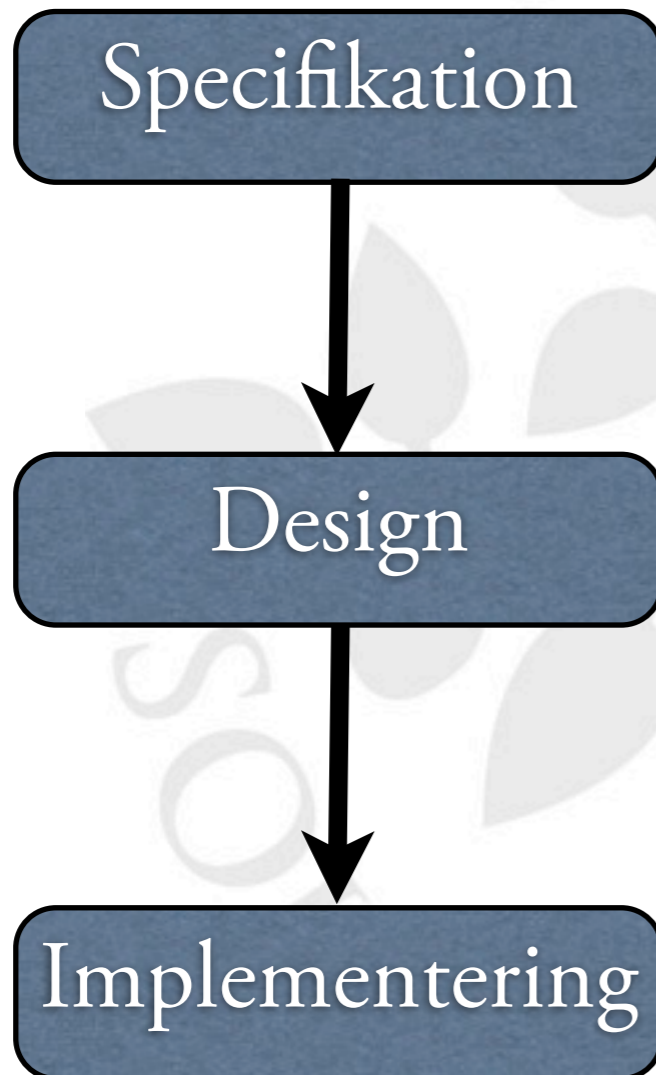
# Abstrakte DataTyper

## Programmering



# Abstrakte DataTyper

## Programmering





# Abstrakte DataTyper

Programmering

ADT

Specifikation

Specifikation

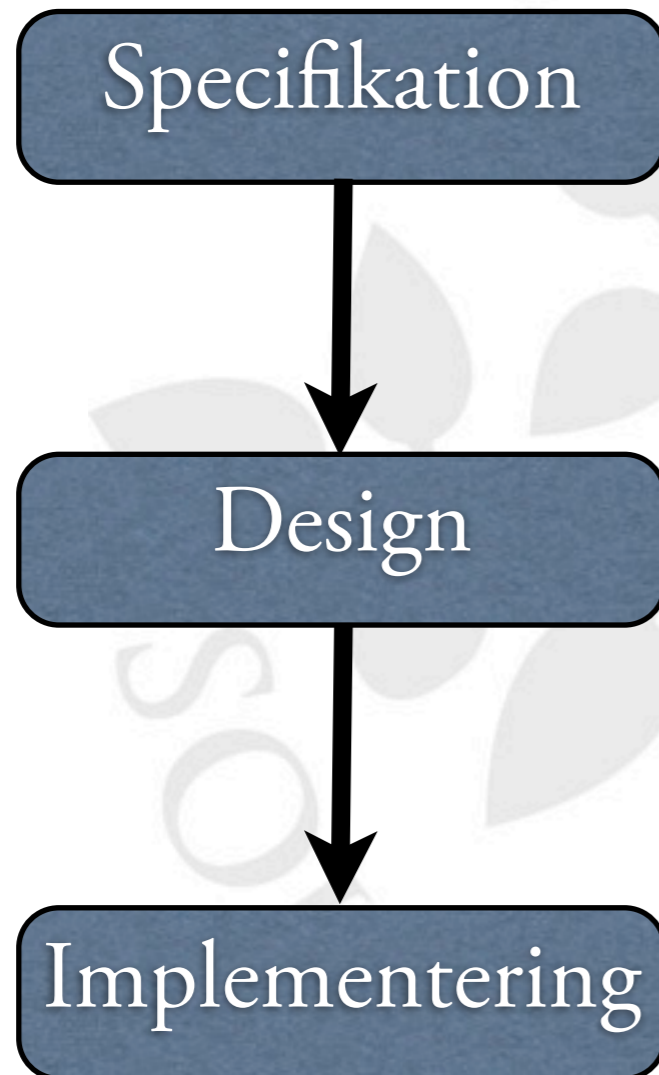
Design

Implementering

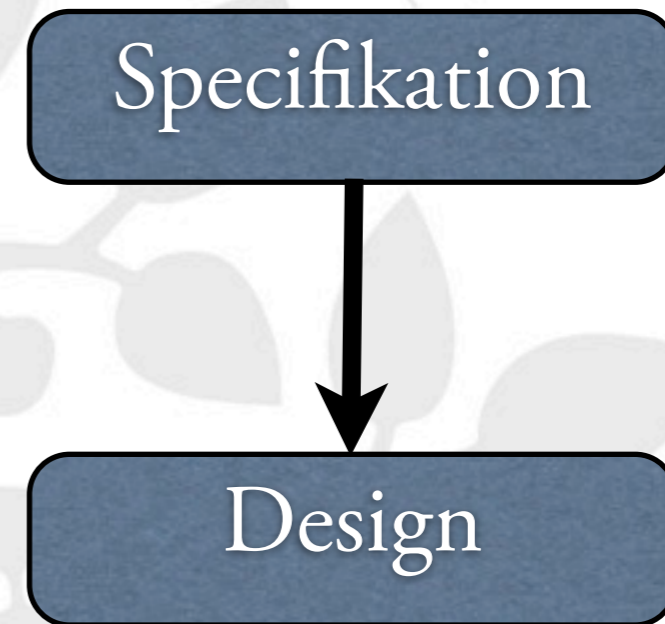


# Abstrakte DataTyper

Programmering

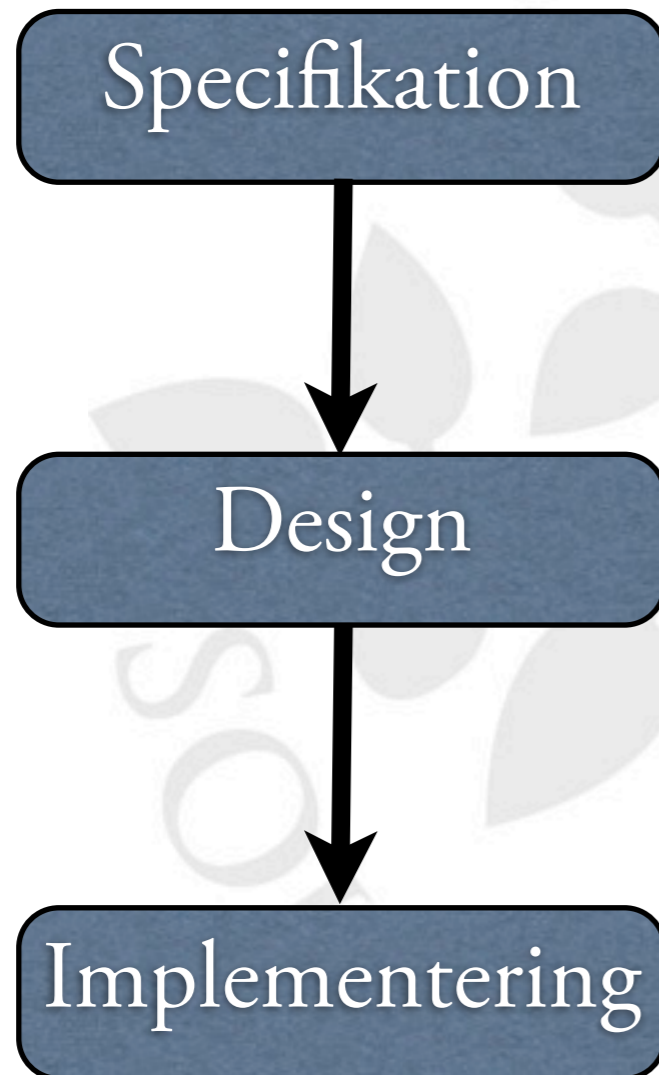


ADT

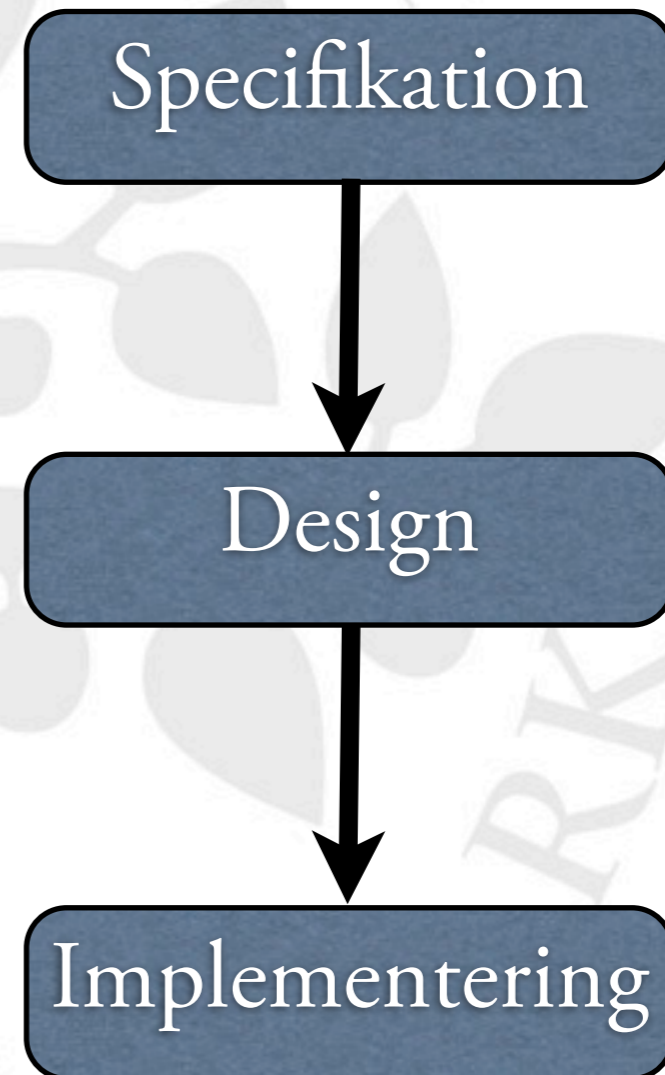


# Abstrakte DataTyper

Programmering



ADT

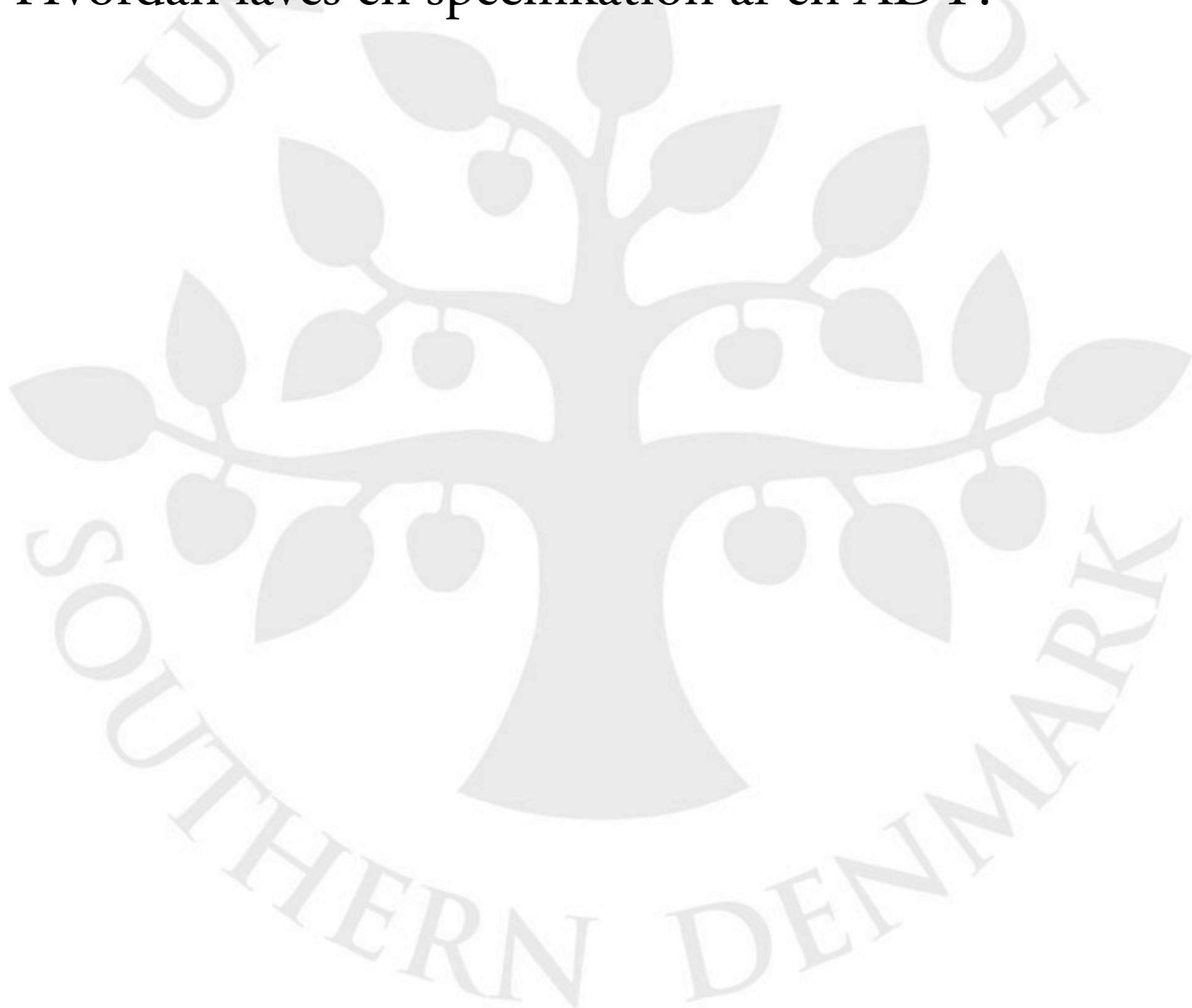


# Abstrakte DataTyper



# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?



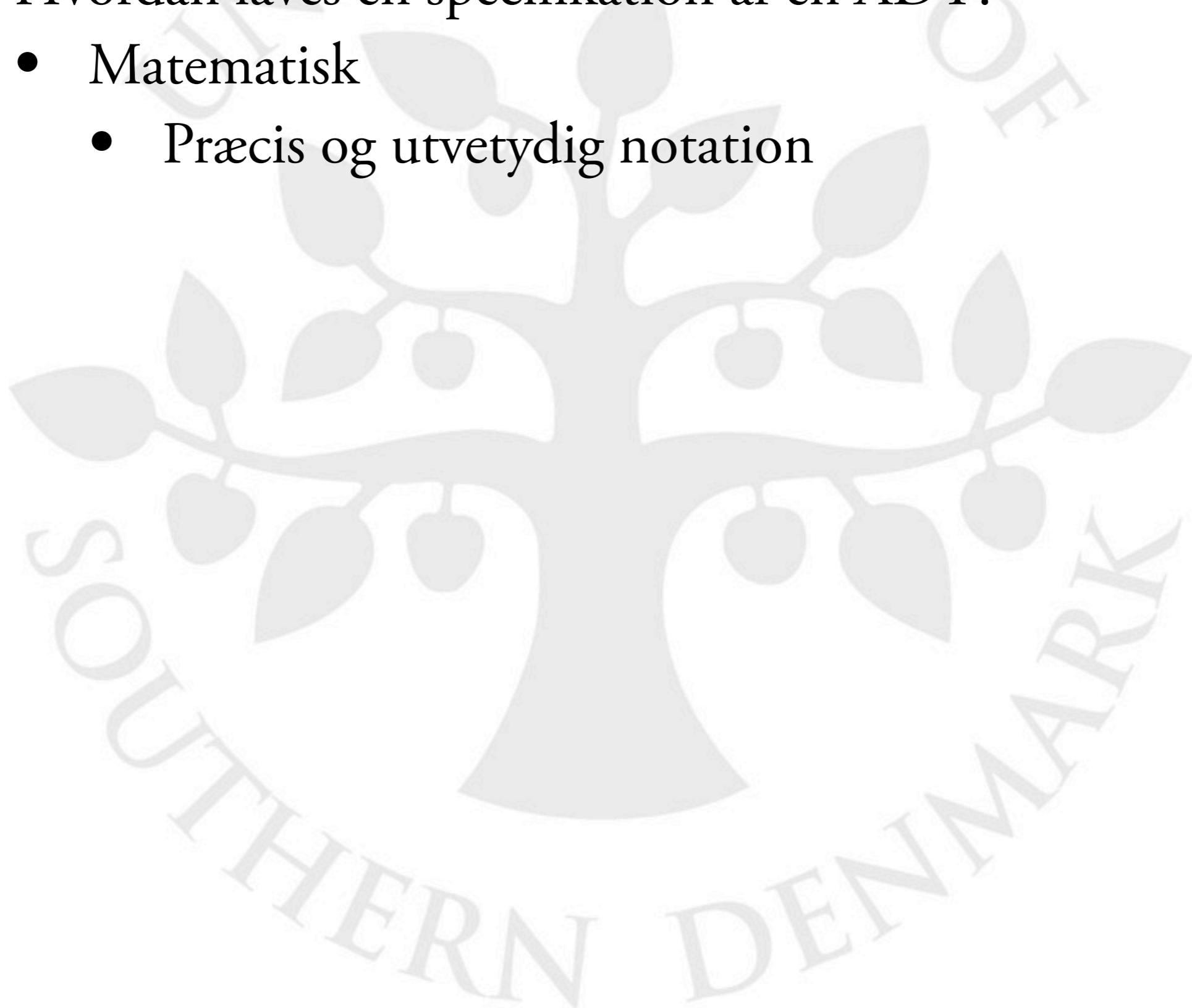
# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk



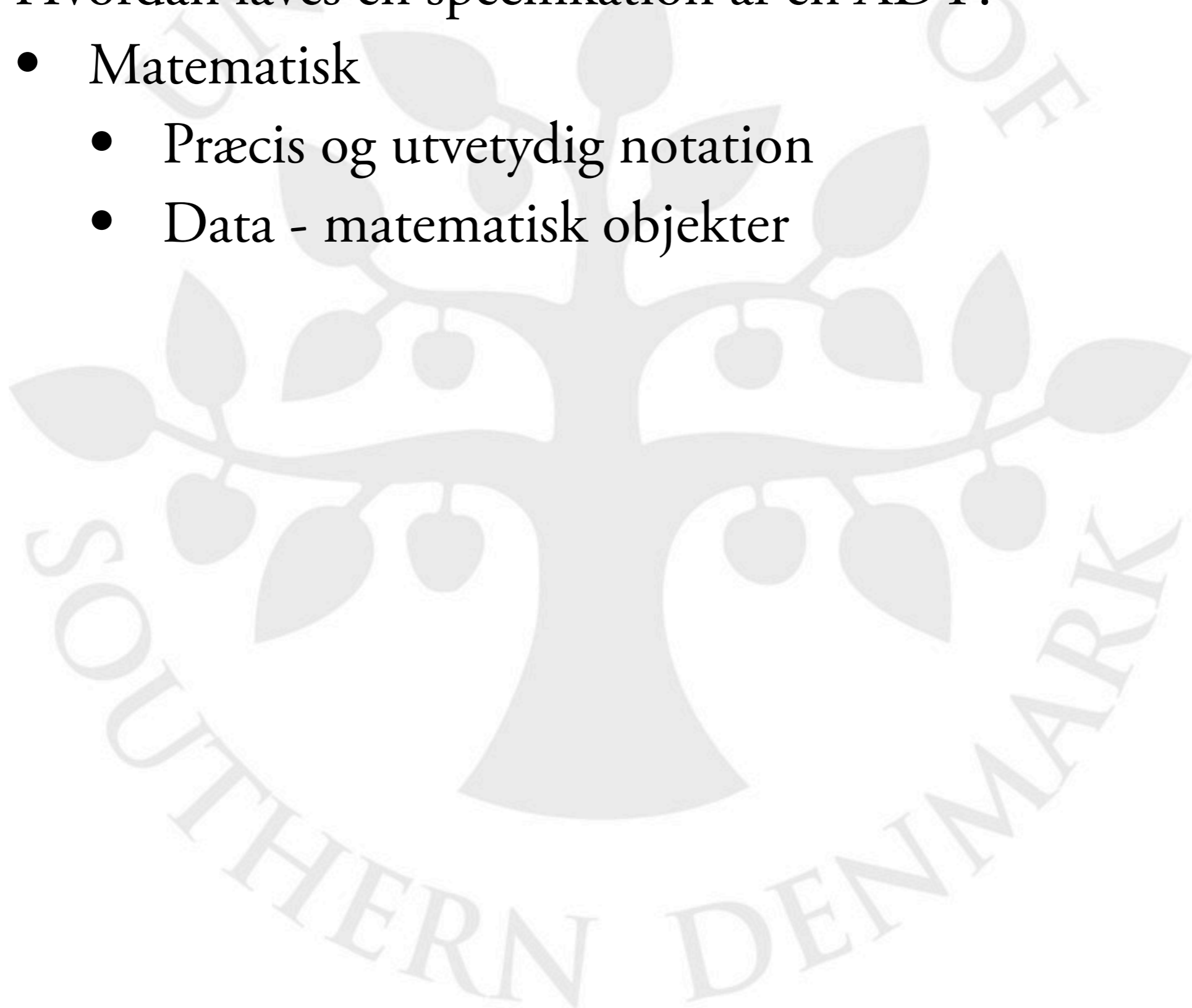
# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation



# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation
    - Data - matematisk objekter





# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation
    - Data - matematisk objekter
      - $\mathbb{N}$ ,  $\mathbb{Z}$ ,  $\mathbb{R}$ , ...



# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation
    - Data - matematisk objekter
      - $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \dots$
      - Objekter med en ordning

# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation
    - Data - matematisk objekter
      - $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \dots$
      - Objekter med en ordning
    - Operationer - matematisk funktioner

# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation
    - Data - matematisk objekter
      - $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \dots$
      - Objekter med en ordning
    - Operationer - matematisk funktioner
      - Eksplicitte definitioner

# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation
    - Data - matematisk objekter
      - $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \dots$
      - Objekter med en ordning
    - Operationer - matematisk funktioner
      - Eksplicitte definitioner
        - $f(x) = x^2$

# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation
    - Data - matematisk objekter
      - $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \dots$
      - Objekter med en ordning
    - Operationer - matematisk funktioner
      - Eksplicitte definitioner
        - $f(x) = x^2$
      - Indirekte definitioner

# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Matematisk
    - Præcis og utvetydig notation
    - Data - matematisk objekter
      - $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \dots$
      - Objekter med en ordning
    - Operationer - matematisk funktioner
      - Eksplicitte definitioner
        - $f(x) = x^2$
      - Indirekte definitioner
        - $\text{sqrt} : x \in \mathbb{R}_+ \mapsto y \in \mathbb{R}_+$   
 $x = y^2 \wedge y \geq 0$

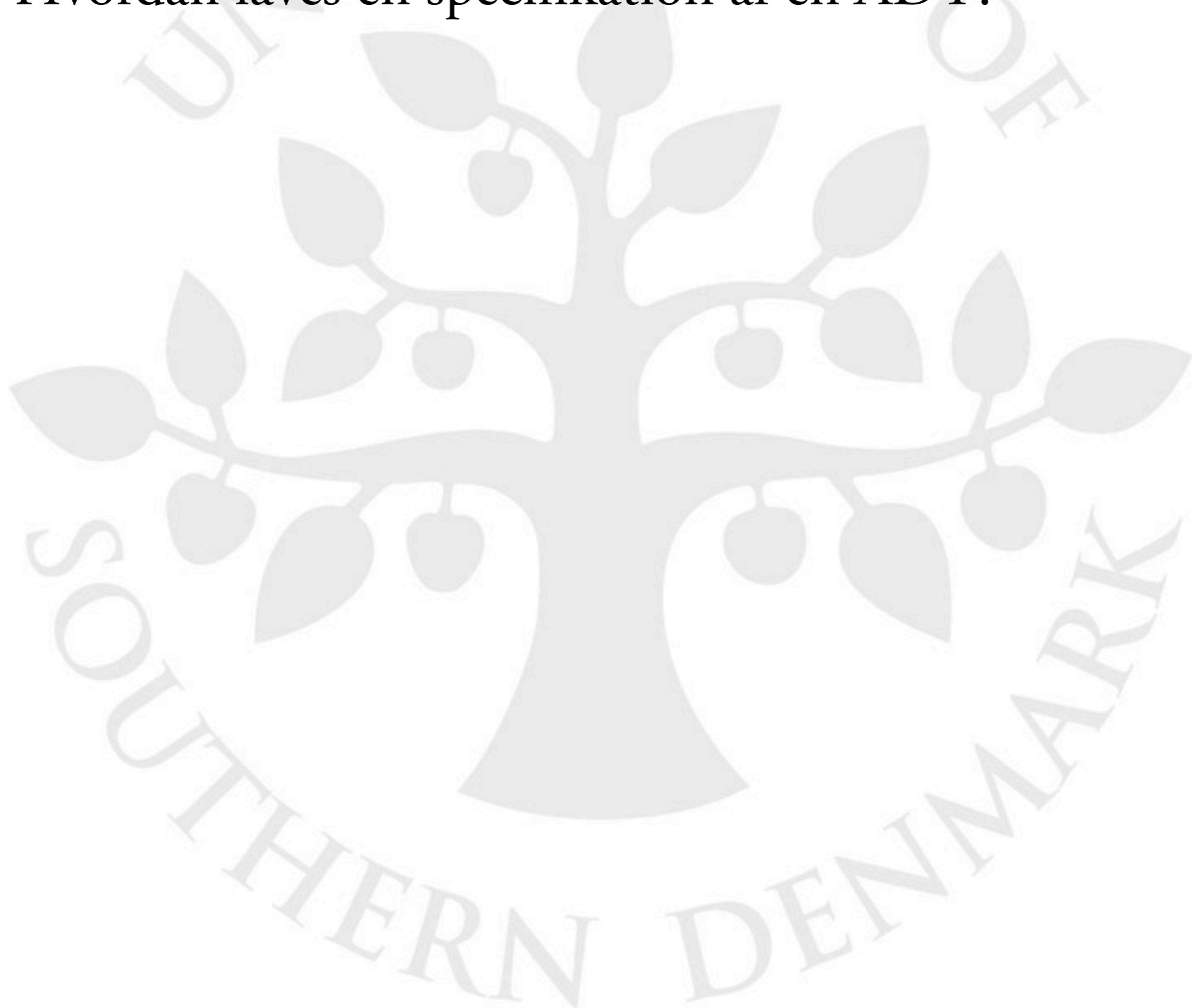
# Abstrakte DataTyper





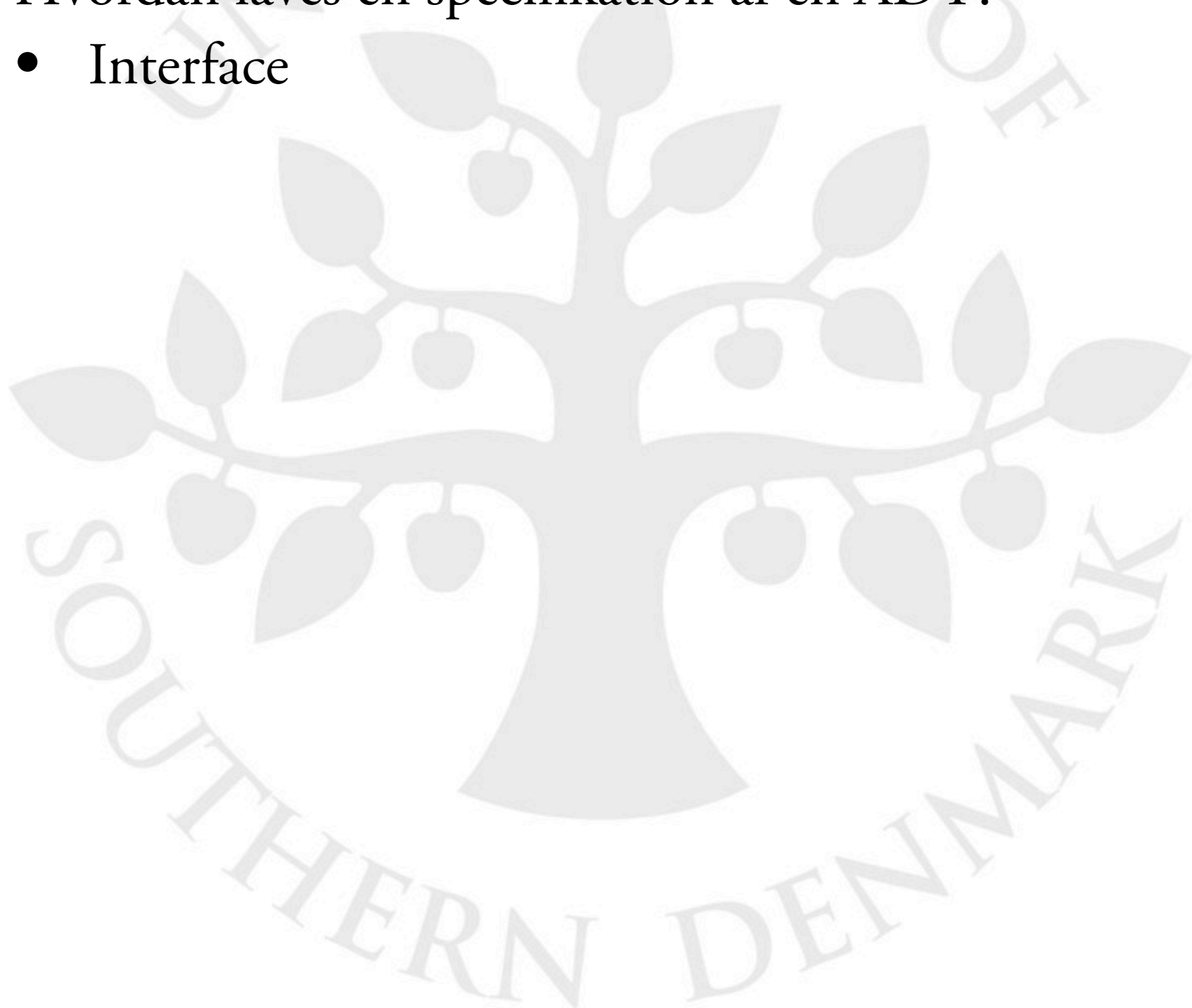
# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?



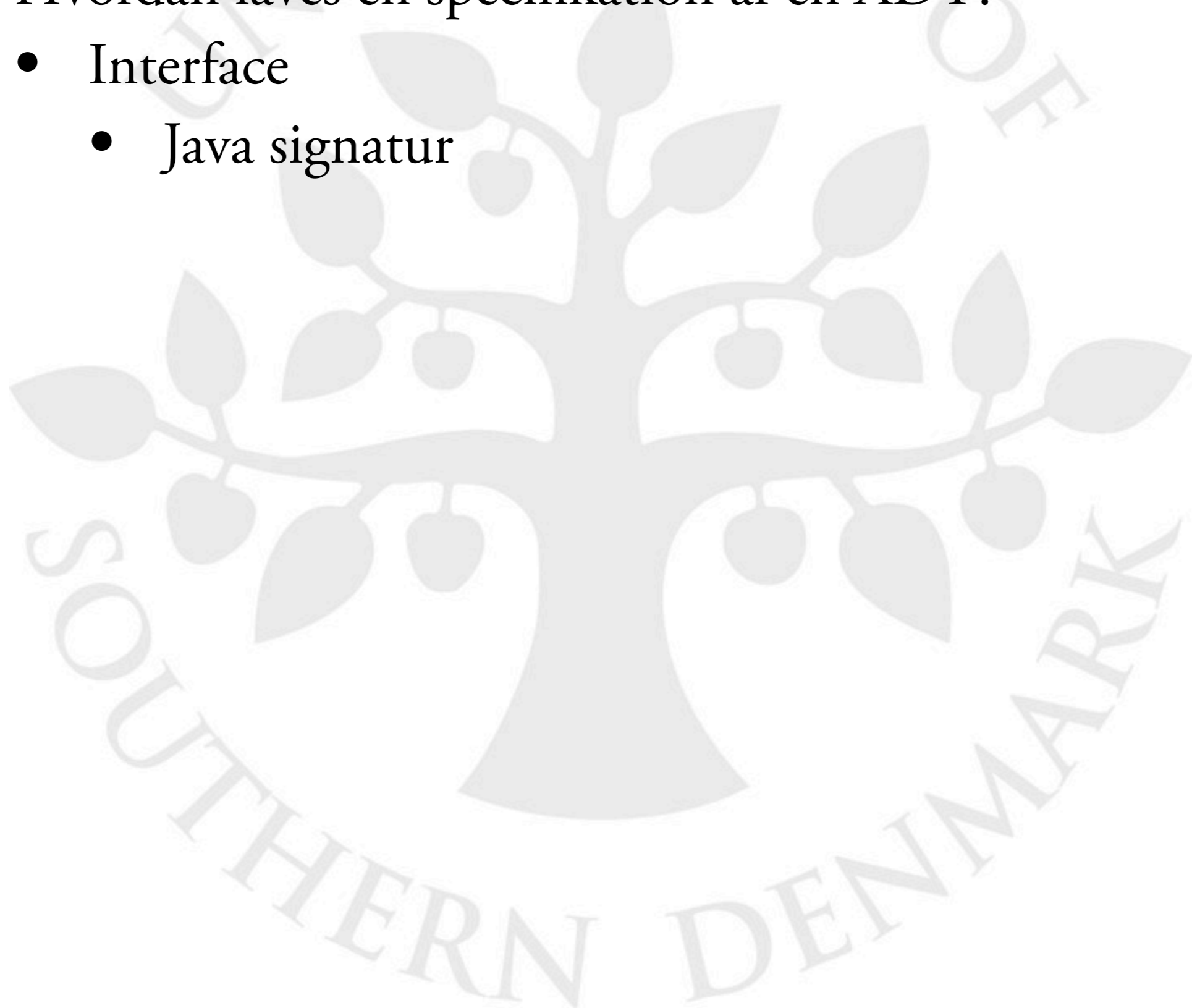
# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Interface



# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Interface
    - Java signatur



# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Interface
    - Java signatur
    - Uformel beskrivelse af hvad metoderne skal gøre



# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Interface
    - Java signatur
    - Uformel beskrivelse af hvad metoderne skal gøre
  - Eksempel



# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Interface
    - Java signatur
    - Uformel beskrivelse af hvad metoderne skal gøre
  - Eksempel
    - ```
public interface Dictionary {  
    public boolean hasWord( String word );  
    public ArrayList<String> suggestions( String word );  
    public void addWord( String word );  
}
```

# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Interface
    - Java signatur
    - Uformel beskrivelse af hvad metoderne skal gøre
  - Eksempel
    - ```
public interface Dictionary {  
    public boolean hasWord( String word );  
    public ArrayList<String> suggestions( String word );  
    public void addWord( String word );  
}
```
    - Sammen med en beskrivelse af hvad hasWord, suggestions og addWord skal gøre

# Abstrakte DataTyper

- Hvordan laves en specifikation af en ADT?
  - Interface
    - Java signatur
    - Uformel beskrivelse af hvad metoderne skal gøre
  - Eksempel
    - ```
public interface Dictionary {  
    public boolean hasWord( String word );  
    public ArrayList<String> suggestions( String word );  
    public void addWord( String word );  
}
```
    - Sammen med en beskrivelse af hvad `hasWord`, `suggestions` og `addWord` skal gøre
      - “`hasWord` slår op i ordbogen og undersøger om det givne ord findes.”



# Abstrakte DataTyper



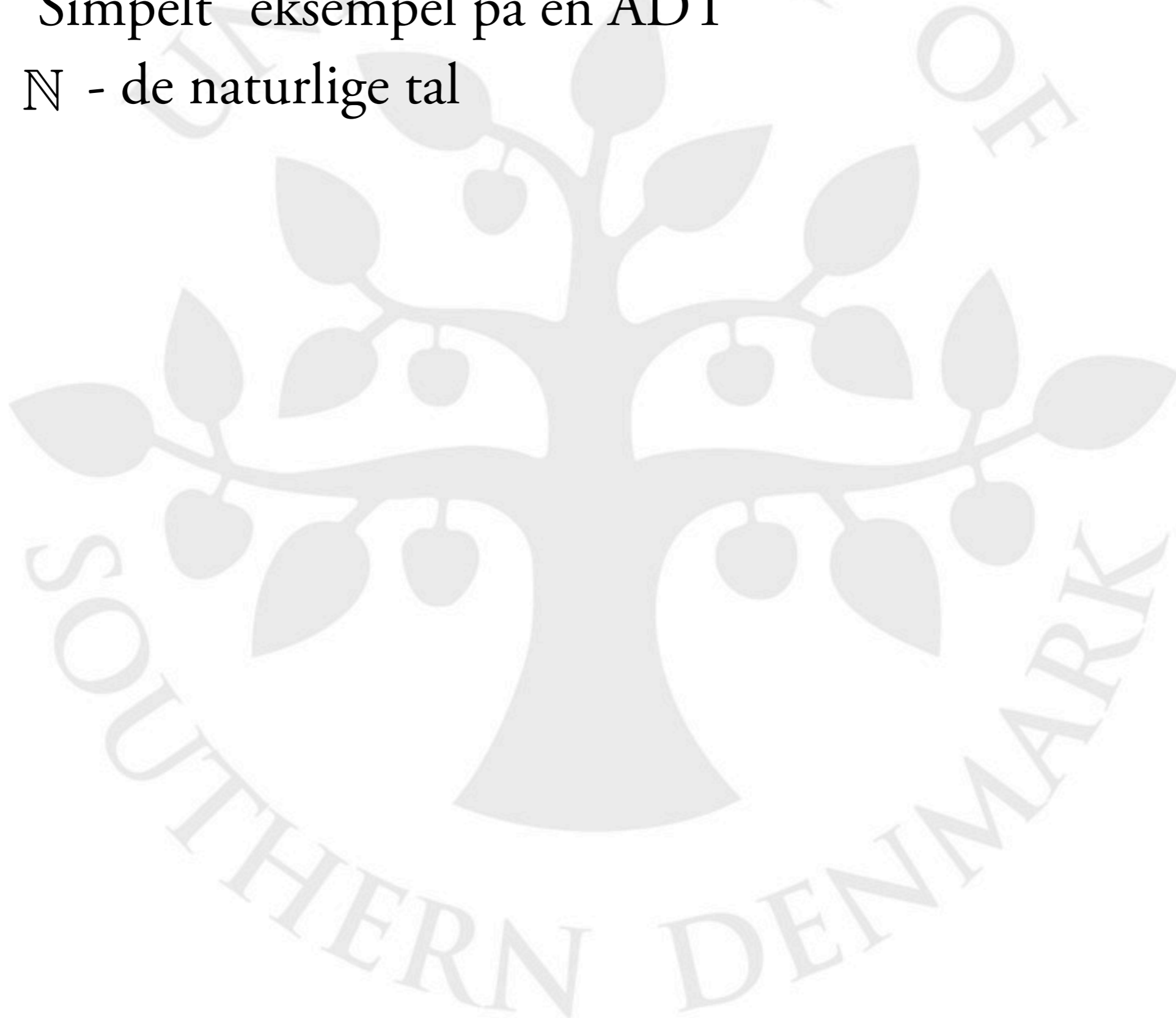
# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT



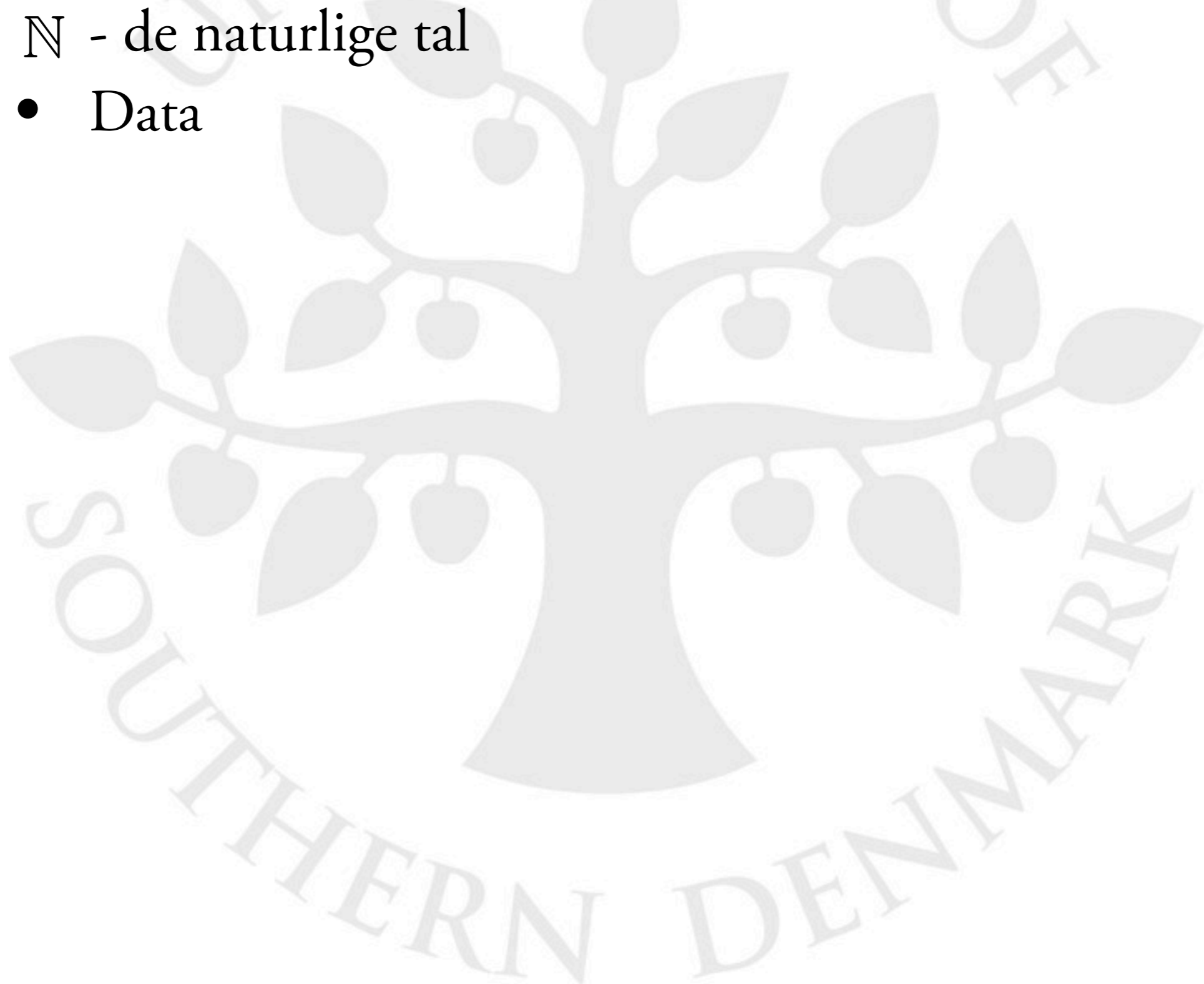
# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal



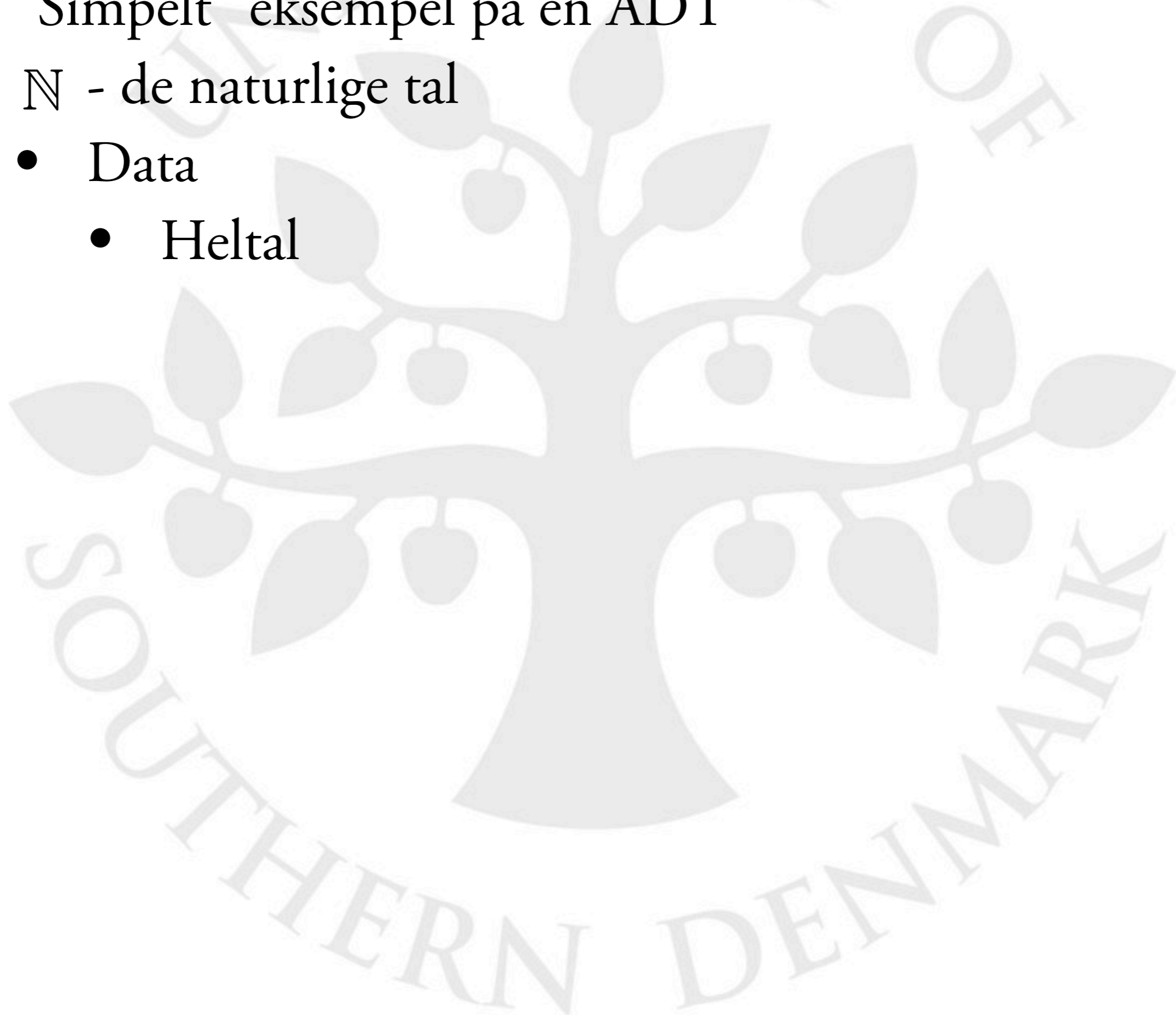
# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal
  - Data



# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal
  - Data
    - Heltal



# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal
  - Data
    - Heltal
    - 0, 1, 2, ...



# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal
  - Data
    - Heltal
    - 0, 1, 2, ...
  - Operationer



# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal
  - Data
    - Heltal
    - 0, 1, 2, ...
  - Operationer
    - Addition





# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal
  - Data
    - Heltal
    - 0, 1, 2, ...
  - Operationer
    - Addition
      - $+ : (x, y) \in (\mathbb{N}, \mathbb{N}) \mapsto z \in \mathbb{N}$   
 $z = x + y$

# Abstrakte DataTyper

- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal
  - Data
    - Heltal
    - 0, 1, 2, ...
  - Operationer
    - Addition
      - $+ : (x, y) \in (\mathbb{N}, \mathbb{N}) \mapsto z \in \mathbb{N}$   
 $z = x + y$
    - Multiplikation

# Abstrakte DataTyper

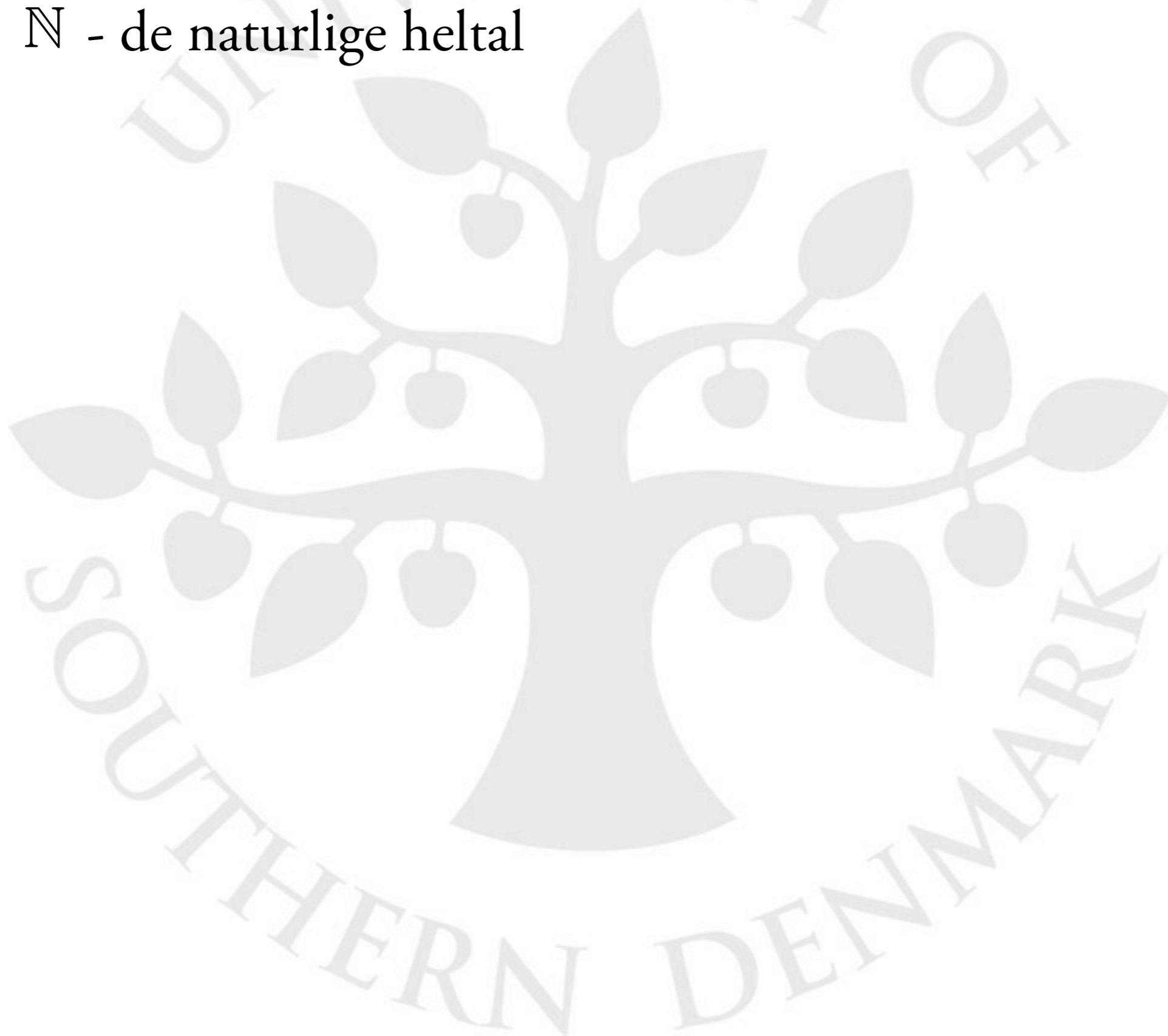
- “Simpelt” eksempel på en ADT
- $\mathbb{N}$  - de naturlige tal
  - Data
    - Heltal
    - 0, 1, 2, ...
  - Operationer
    - Addition
      - $+ : (x, y) \in (\mathbb{N}, \mathbb{N}) \mapsto z \in \mathbb{N}$   
 $z = x + y$
    - Multiplikation
    - ...

# Abstrakte DataTyper



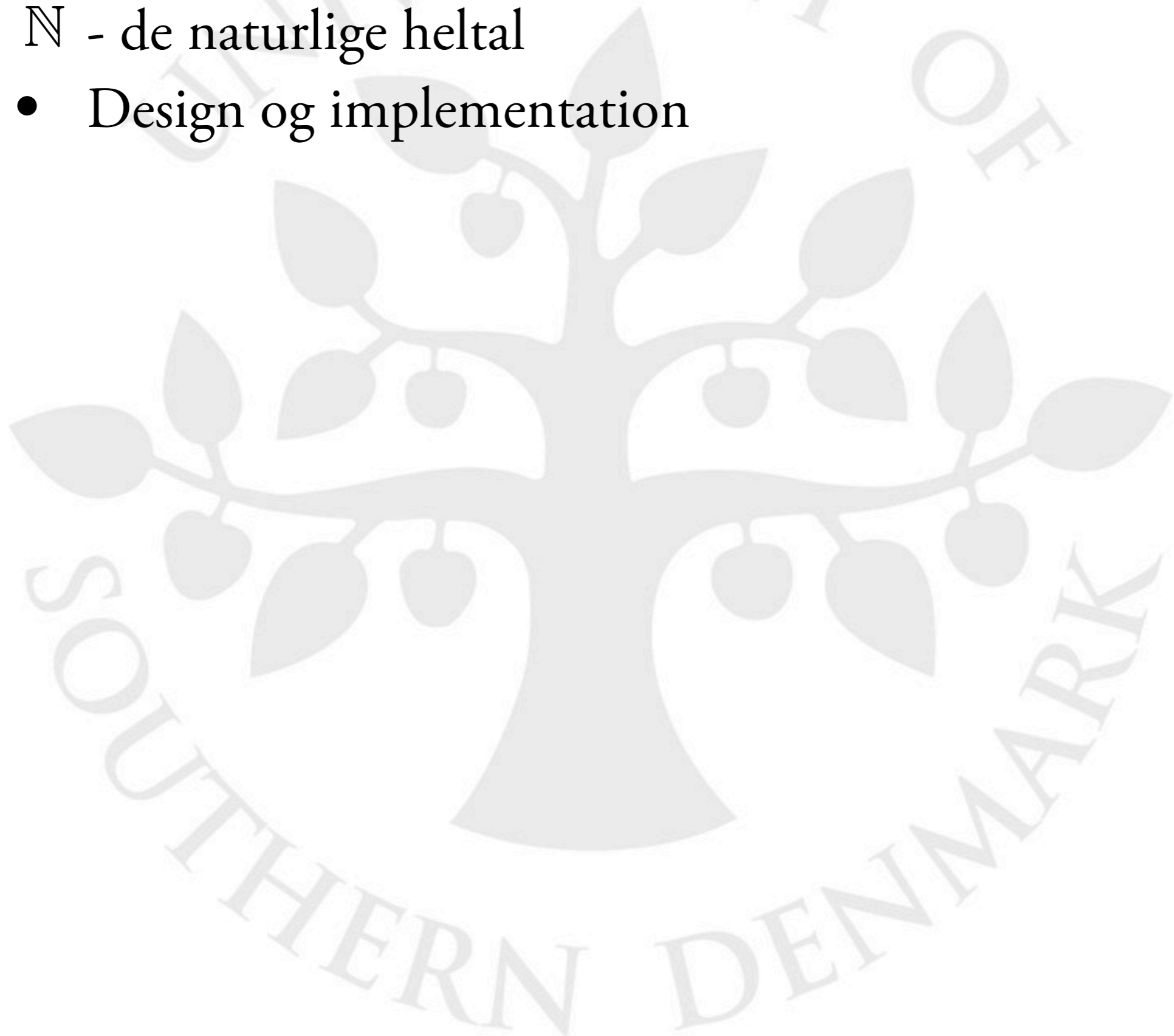
# Abstrakte DataTyper

- $\mathbb{N}$  - de naturlige heltal



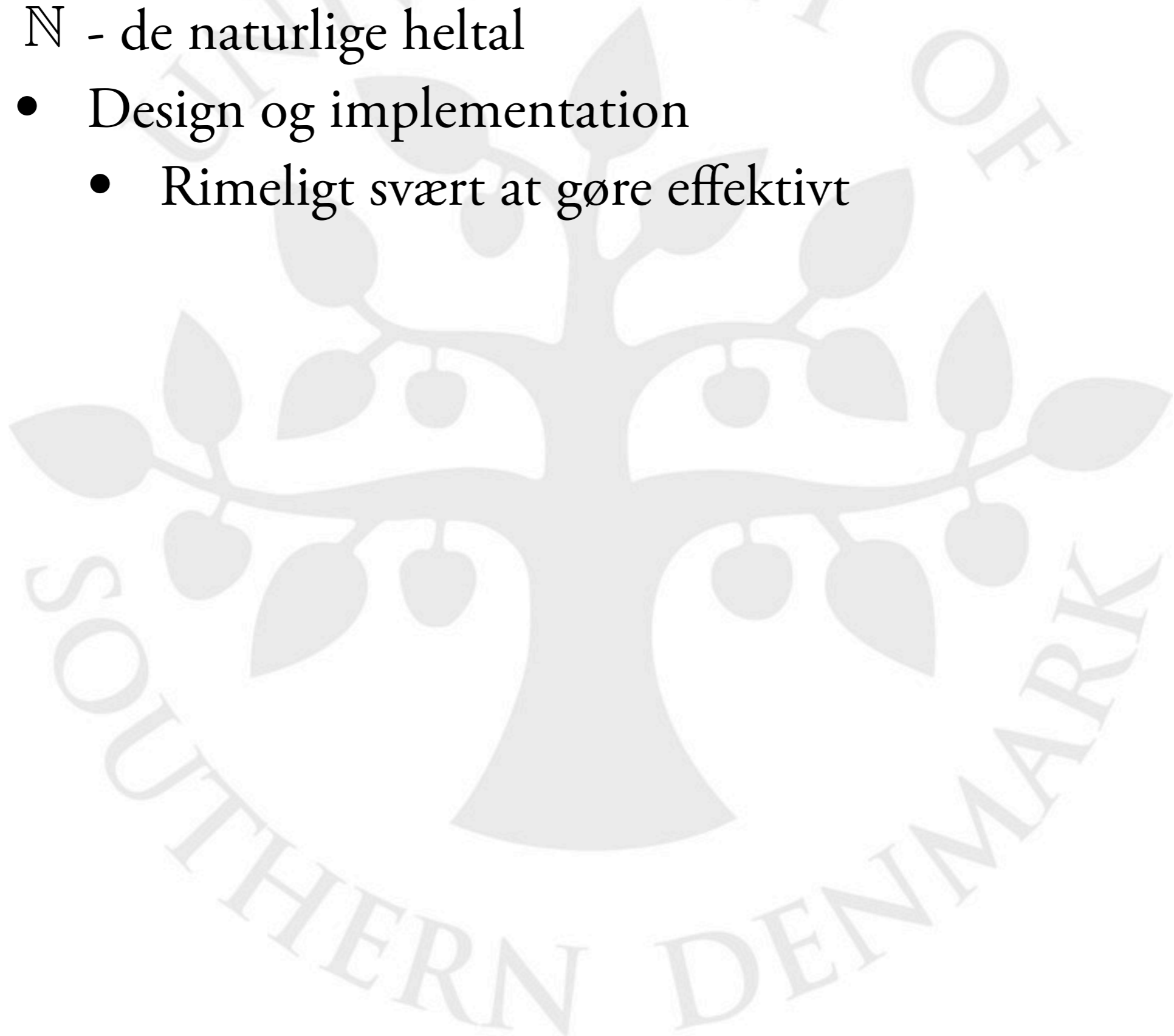
# Abstrakte DataTyper

- $\mathbb{N}$  - de naturlige heltal
- Design og implementation



# Abstrakte DataTyper

- $\mathbb{N}$  - de naturlige heltal
- Design og implementation
  - Rimeligt svært at gøre effektivt



# Abstrakte DataTyper

- $\mathbb{N}$  - de naturlige heltal
- Design og implementation
  - Rimeligt svært at gøre effektivt
  - DM506 - Maskinarkitektur





# Abstrakte DataTyper

- $\mathbb{N}$  - de naturlige heltal
- Design og implementation
  - Rimeligt svært at gøre effektivt
  - DM506 - Maskinarkitektur
- Tallene i Java



# Abstrakte DataTyper

- $\mathbb{N}$  - de naturlige heltal
- Design og implementation
  - Rimeligt svært at gøre effektivt
  - DM506 - Maskinarkitektur
- Tallene i Java
  - Nogen har gjort arbejdet for os



# Abstrakte DataTyper

- $\mathbb{N}$  - de naturlige heltal
- Design og implementation
  - Rimeligt svært at gøre effektivt
  - DM506 - Maskinarkitektur
- Tallene i Java
  - Nogen har gjort arbejdet for os
  - `int`, `double`, ...

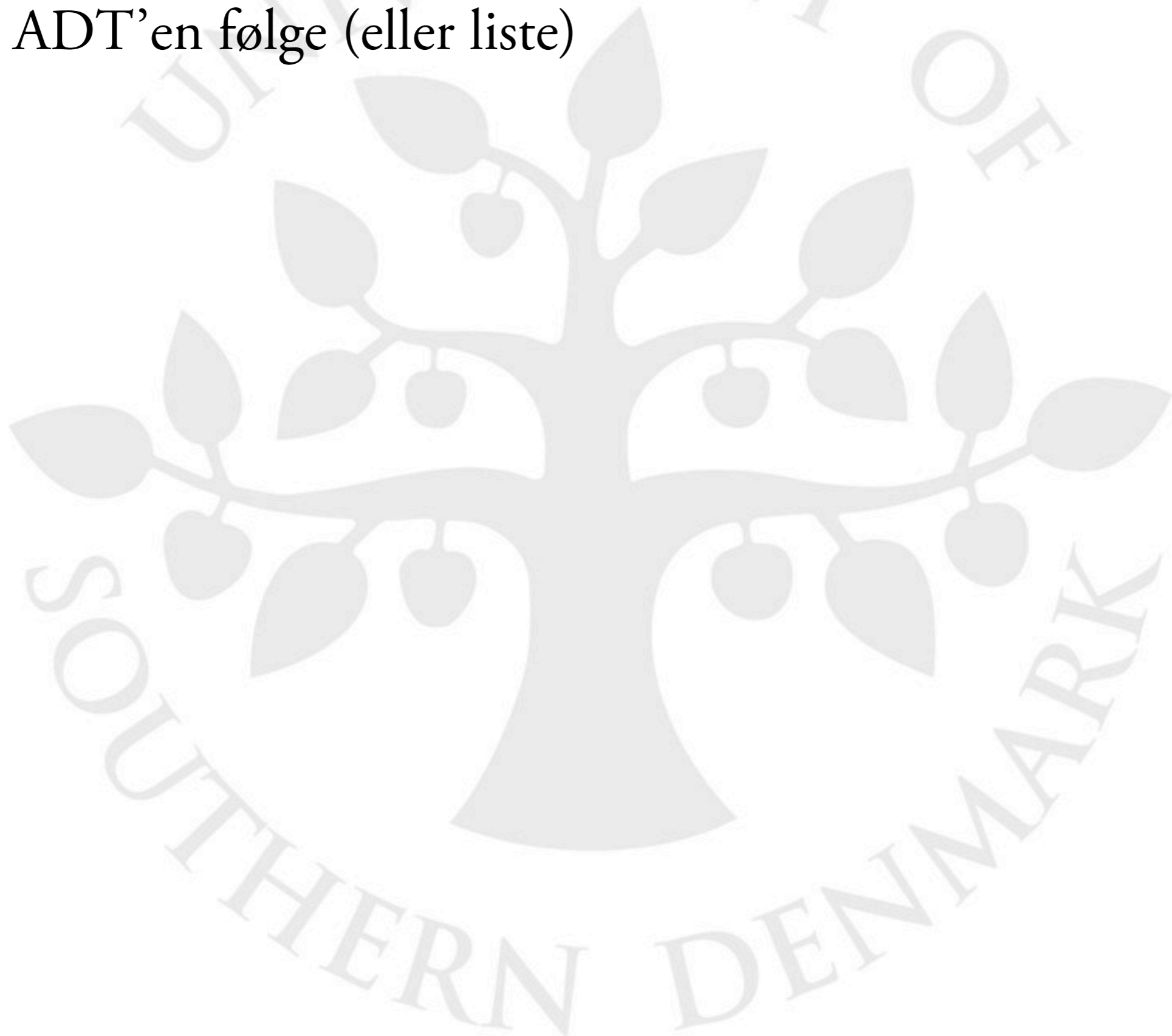
# Lister





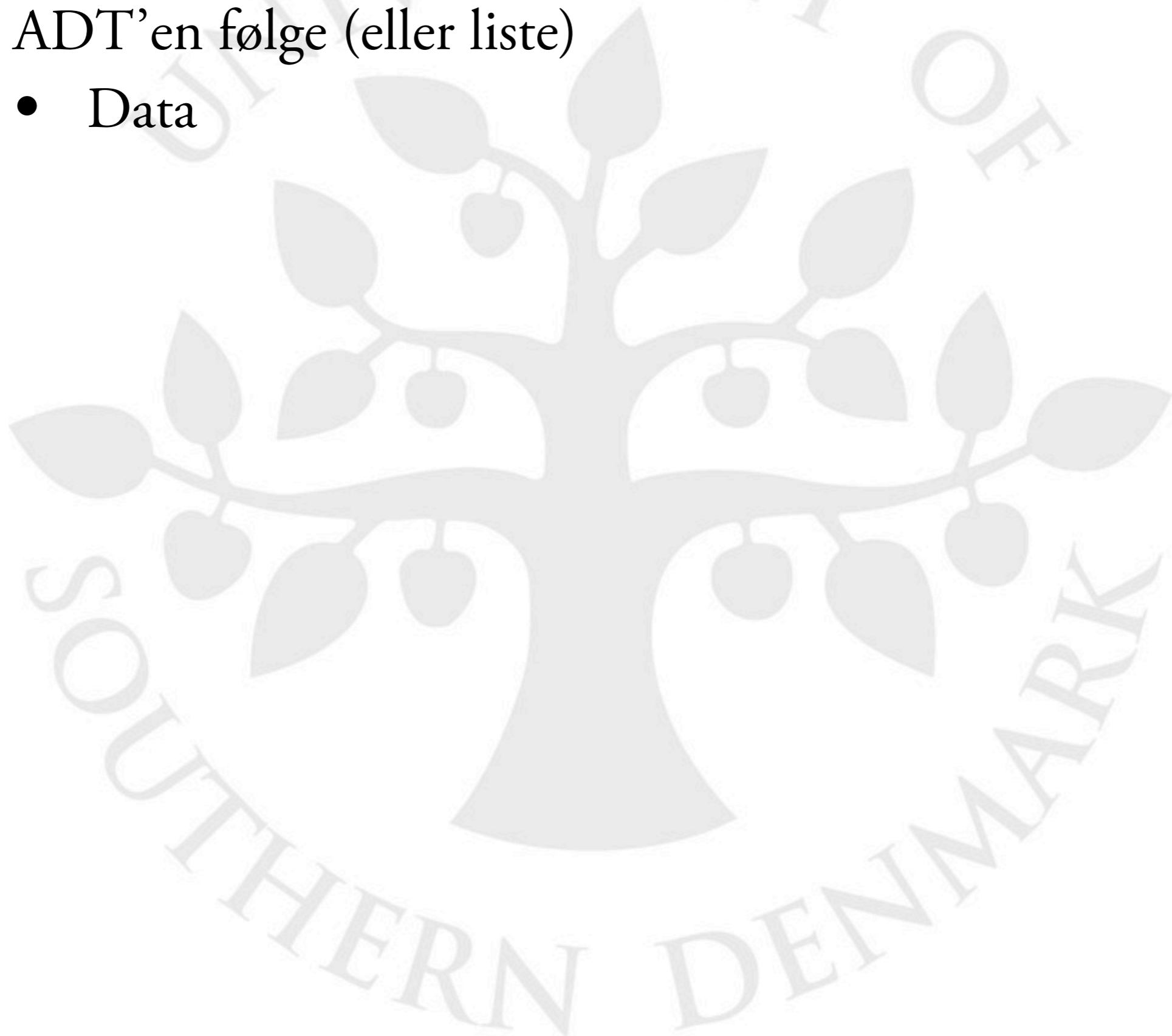
# Lister

- ADT'en følge (eller liste)



# Lister

- ADT'en følge (eller liste)
  - Data





# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner



# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer





# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - `add(e)`



# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - add(e)
    - get(i)



# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - add(e)
    - get(i)
    - set(i, e)



# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - add(e)
    - get(i)
    - set(i, e)
    - remove(i)



# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - add(e)
    - get(i)
    - set(i, e)
    - remove(i)
    - size()

# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - add(e)
    - get(i)
    - set(i, e)
    - remove(i)
    - size()
    - subList(i, j)

# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - add(e)
    - get(i)
    - set(i, e)
    - remove(i)
    - size()
    - subList(i, j)
    - ?

# Lister

- ADT'en følge (eller liste)
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - add(e)
    - get(i)
    - set(i, e)
    - remove(i)
    - size()
    - subList(i, j)
    - ?
- Se `java.util.List` interfacet





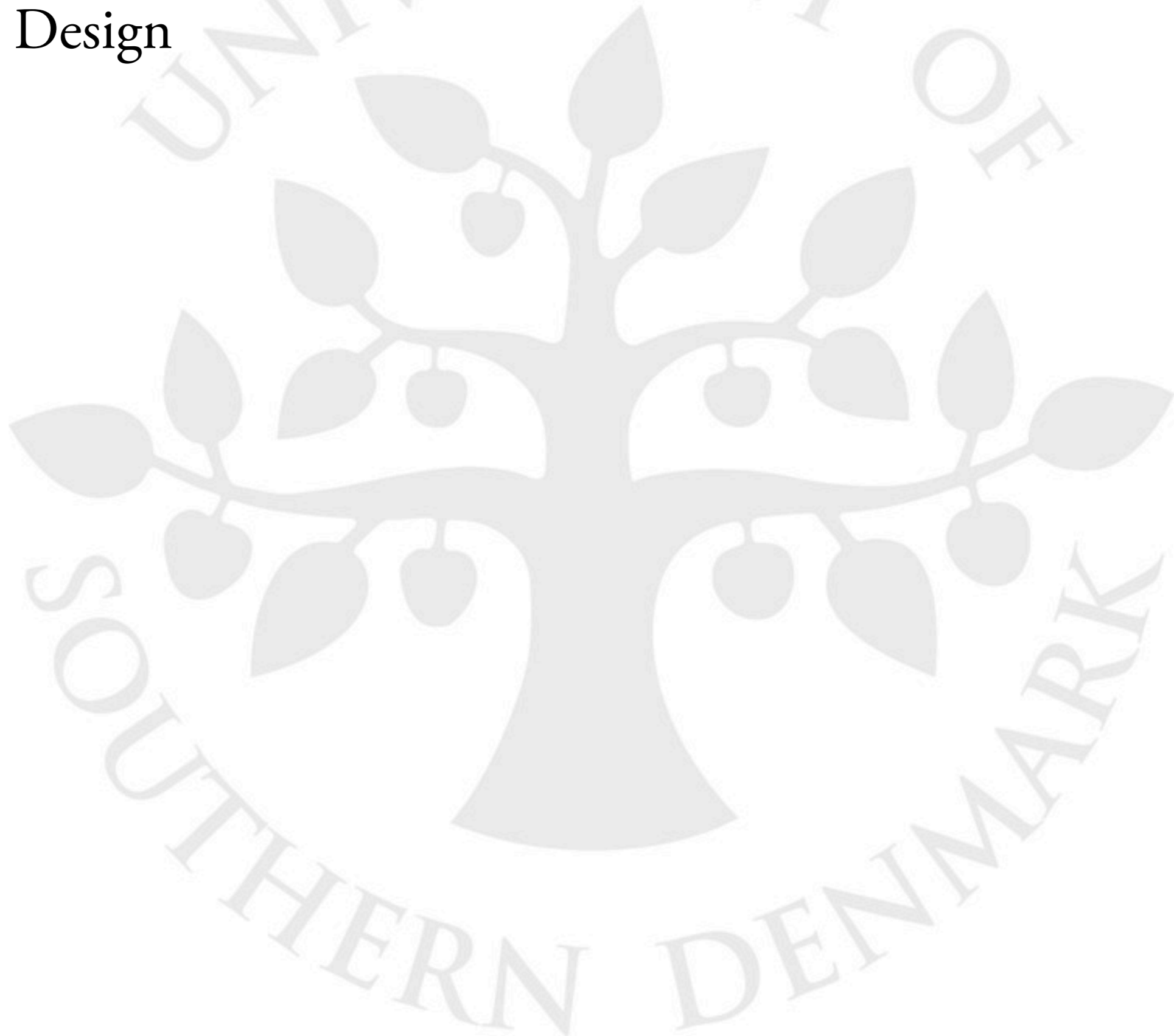
# Lister





# Lister

- Design



# Lister

- Design
  - Ikke alle operationer kan være effektive





# Lister

- Design
  - Ikke alle operationer kan være effektive
    - `add(e)` og `get(i)` er i “konflikt”



# Lister

- Design
  - Ikke alle operationer kan være effektive
    - `add(e)` og `get(i)` er i “konflikt”
  - Tre muligheder



# Lister

- Design
  - Ikke alle operationer kan være effektive
    - `add(e)` og `get(i)` er i “konflikt”
  - Tre muligheder
    - Array



# Lister

- Design
  - Ikke alle operationer kan være effektive
    - `add(e)` og `get(i)` er i “konflikt”
  - Tre muligheder
    - Array
    - Hægtet liste



# Lister

- Design
  - Ikke alle operationer kan være effektive
    - `add(e)` og `get(i)` er i “konflikt”
  - Tre muligheder
    - Array
    - Hægtet liste
    - Dynamisk array





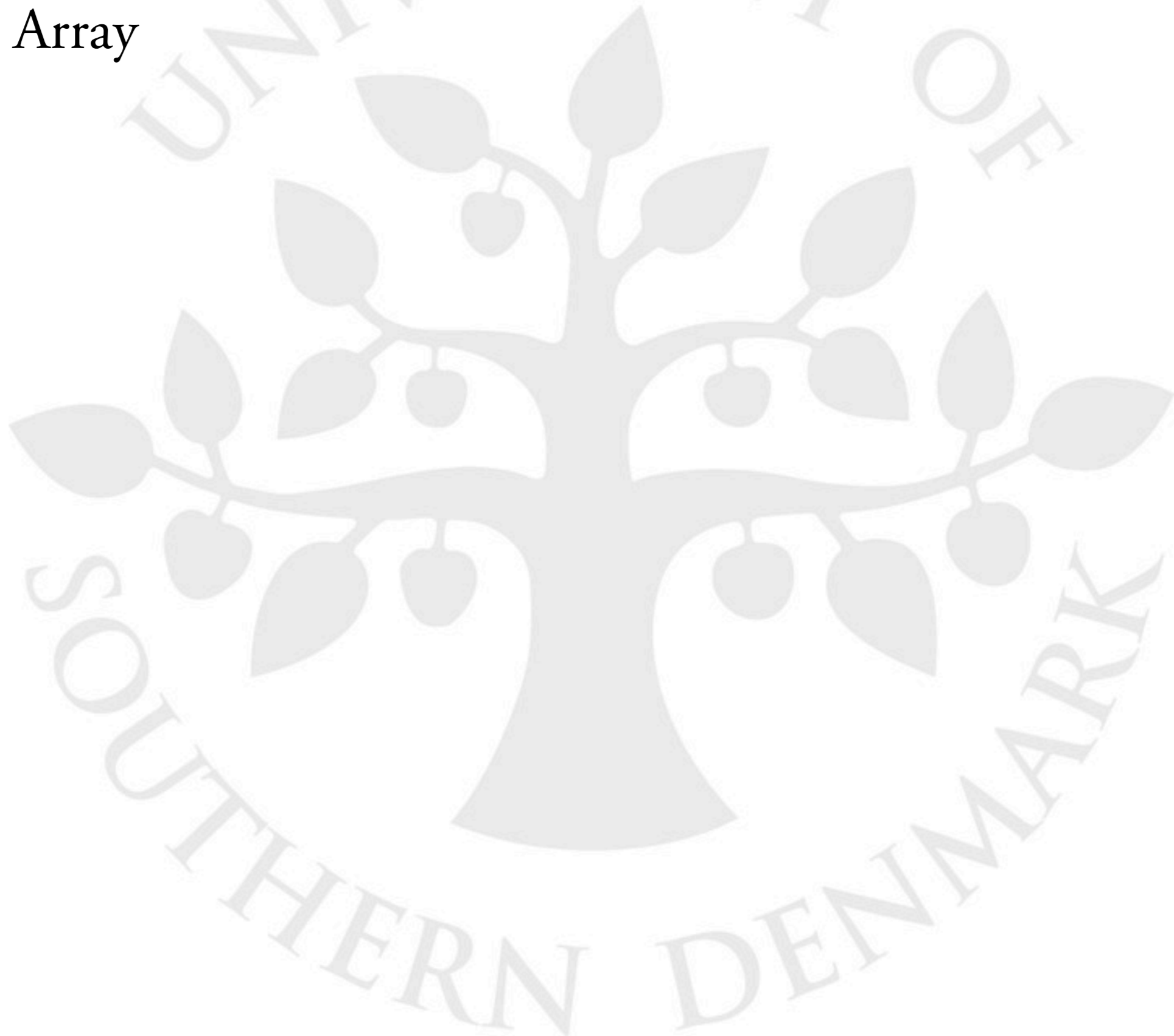
# Lister





# Lister

- Array





# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller





# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis





# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”

# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”
    - Kan ikke indsætte på simpel vis midt i listen



# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”
    - Kan ikke indsætte på simpel vis midt i listen
  - Alle andre operationer er effektive



# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”
    - Kan ikke indsætte på simpel vis midt i listen
  - Alle andre operationer er effektive
    - `get(i)`, `set(i, e)`





# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”
    - Kan ikke indsætte på simpel vis midt i listen
  - Alle andre operationer er effektive
    - `get(i)`, `set(i, e)`
    - Kan regne plads i hukommelsen ud meget hurtigt

# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”
    - Kan ikke indsætte på simpel vis midt i listen
  - Alle andre operationer er effektive
    - `get(i)`, `set(i, e)`
    - Kan regne plads i hukommelsen ud meget hurtigt
      - Vi kender adressen på begyndelsen af listen (`a`)

# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”
    - Kan ikke indsætte på simpel vis midt i listen
  - Alle andre operationer er effektive
    - `get(i)`, `set(i, e)`
    - Kan regne plads i hukommelsen ud meget hurtigt
      - Vi kender adressen på begyndelsen af listen (a)
      - Vi kender størrelsen af hver celle (b)

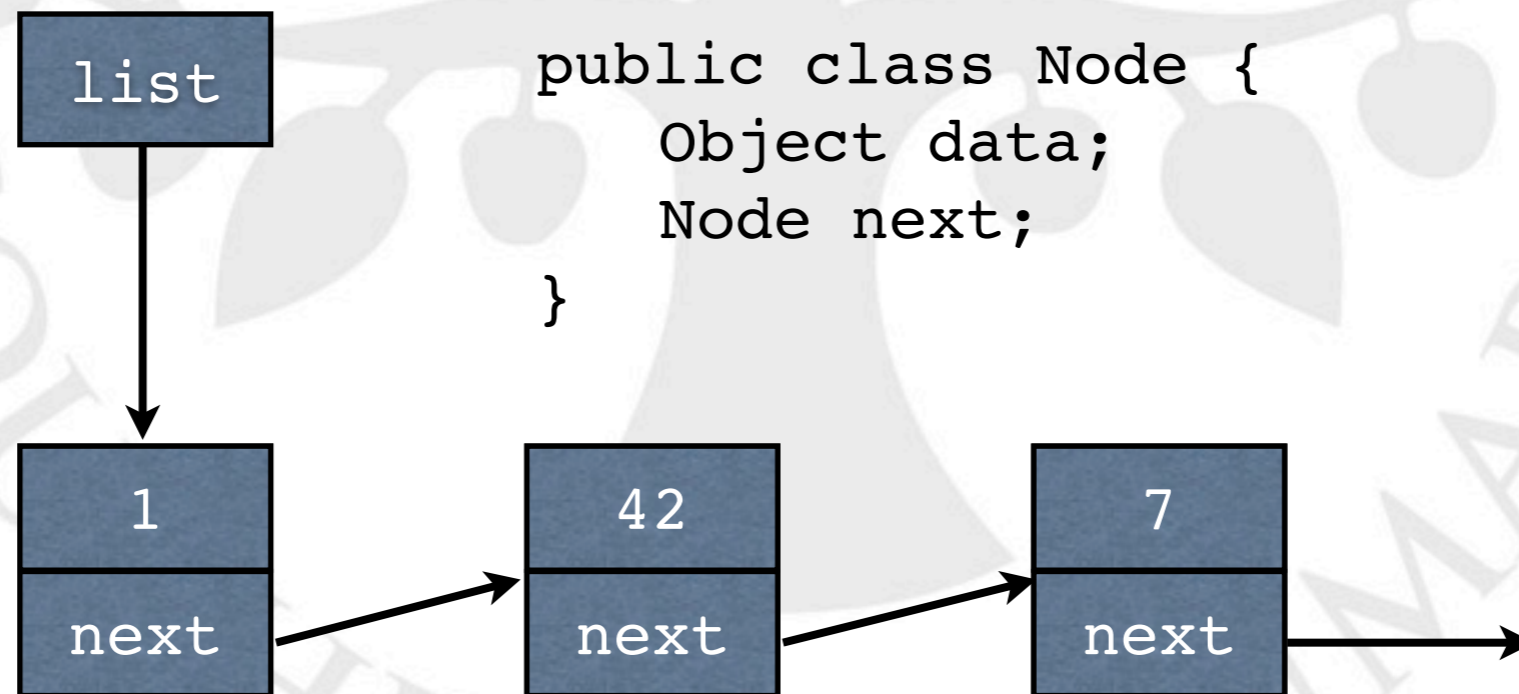
# Lister

- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”
    - Kan ikke indsætte på simpel vis midt i listen
  - Alle andre operationer er effektive
    - $\text{get}(i)$ ,  $\text{set}(i, e)$
    - Kan regne plads i hukommelsen ud meget hurtigt
      - Vi kender adressen på begyndelsen af listen (a)
      - Vi kender størrelsen af hver celle (b)
      - “Plads i” =  $a + b * i$

# Lister

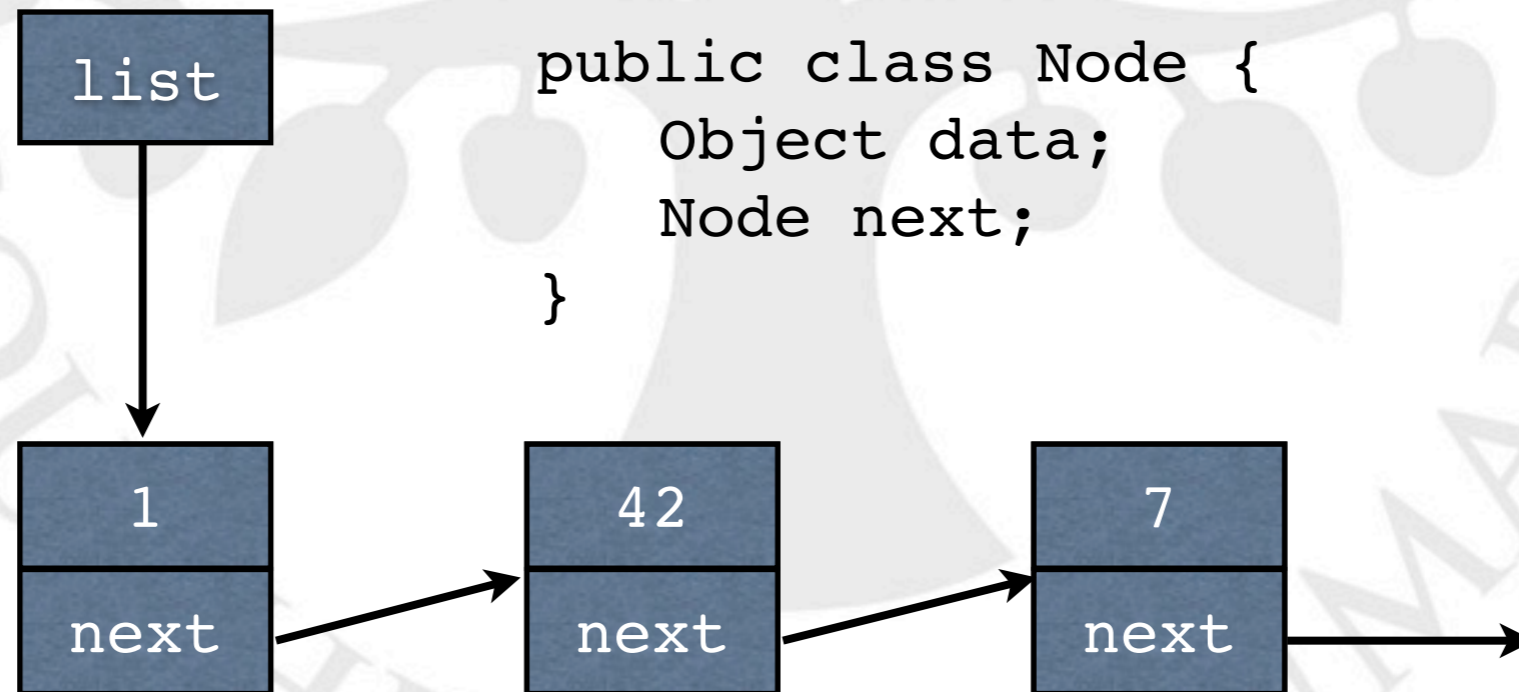
- Array
  - En kontinuerlig sekvens af hukommelsesceller
  - Kompromis
    - Kan ikke indsætte ud over “enden”
    - Kan ikke indsætte på simpel vis midt i listen
  - Alle andre operationer er effektive
    - `get(i)`, `set(i, e)`
    - Kan regne plads i hukommelsen ud meget hurtigt
      - Vi kender adressen på begyndelsen af listen (a)
      - Vi kender størrelsen af hver celle (b)
      - “Plads i” =  $a + b * i$
- Implementering: Javas indbyggede array type

# Lister



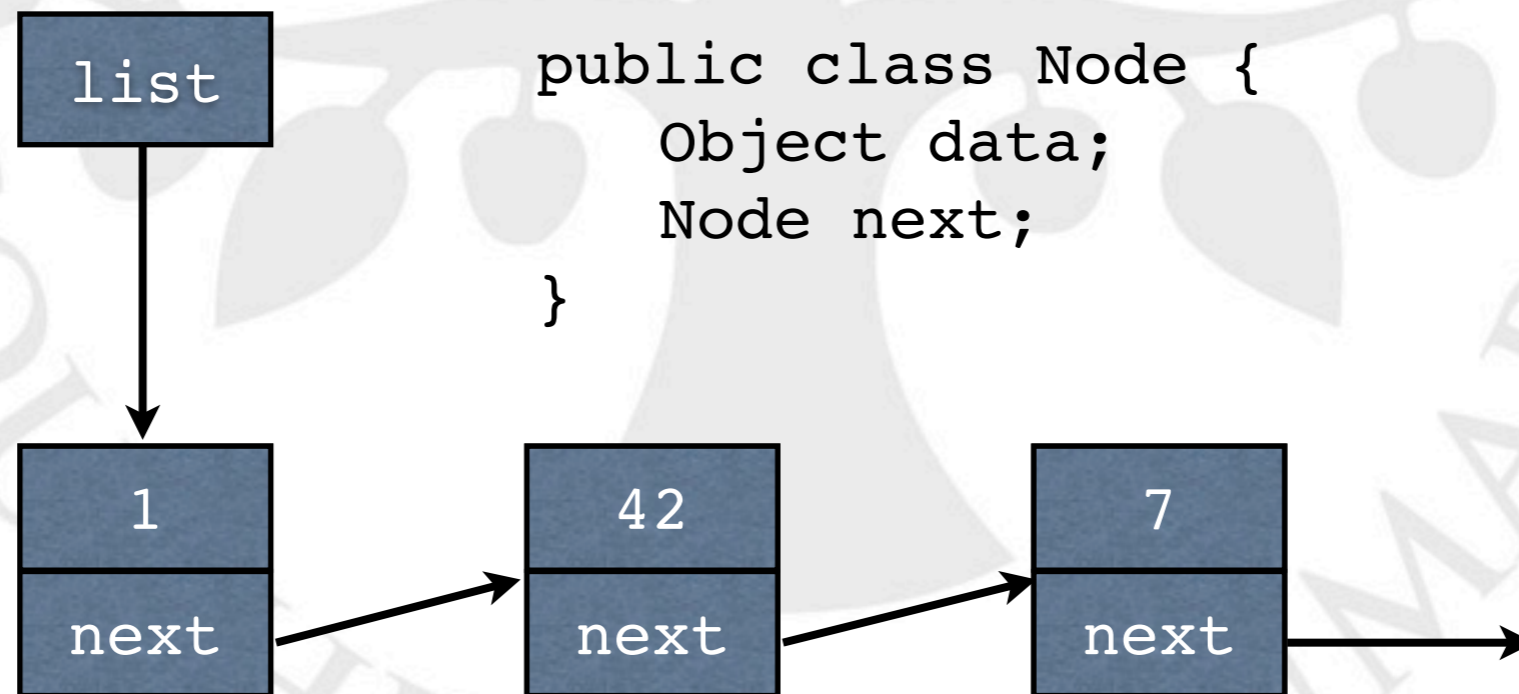
# Lister

- Enkelt-hægtet liste



# Lister

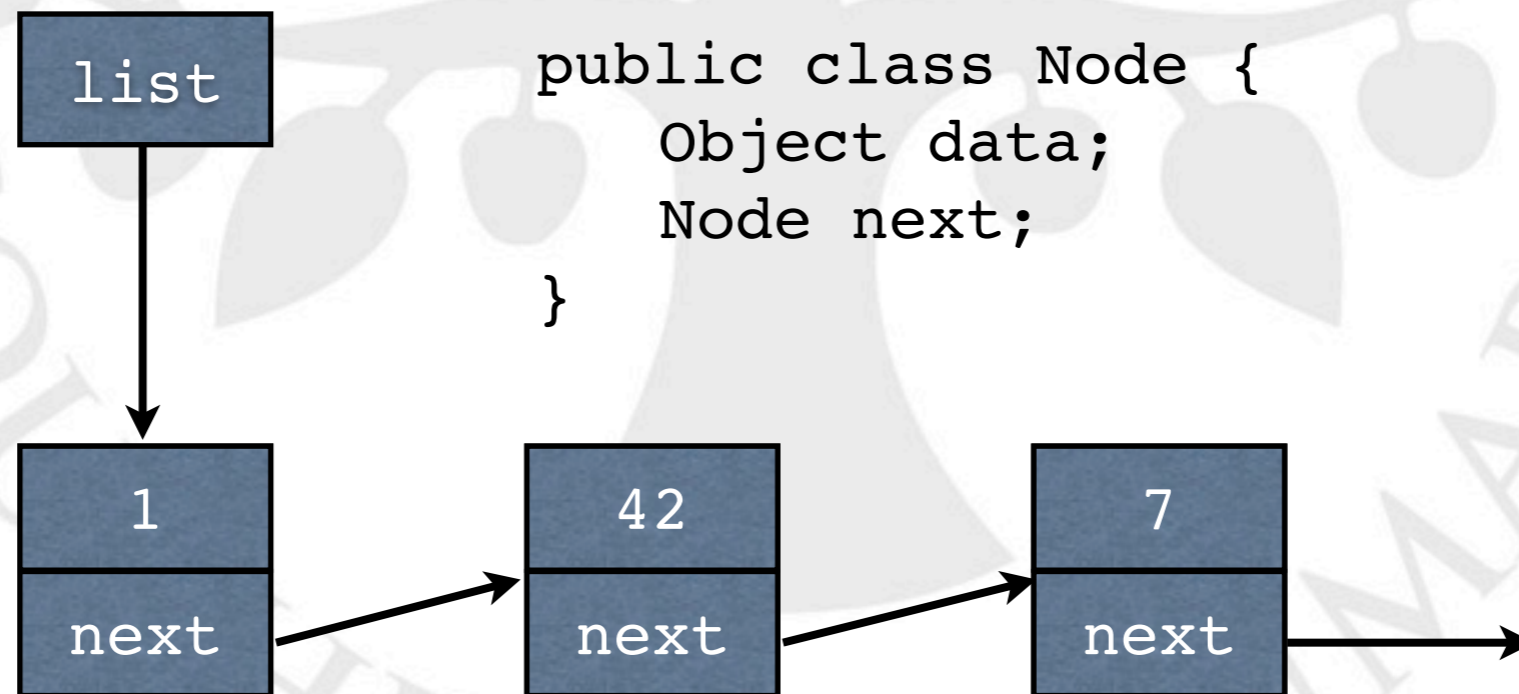
- Enkelt-hægtet liste
  - Knuderne kan ligge hvor som helst i hukommelsen





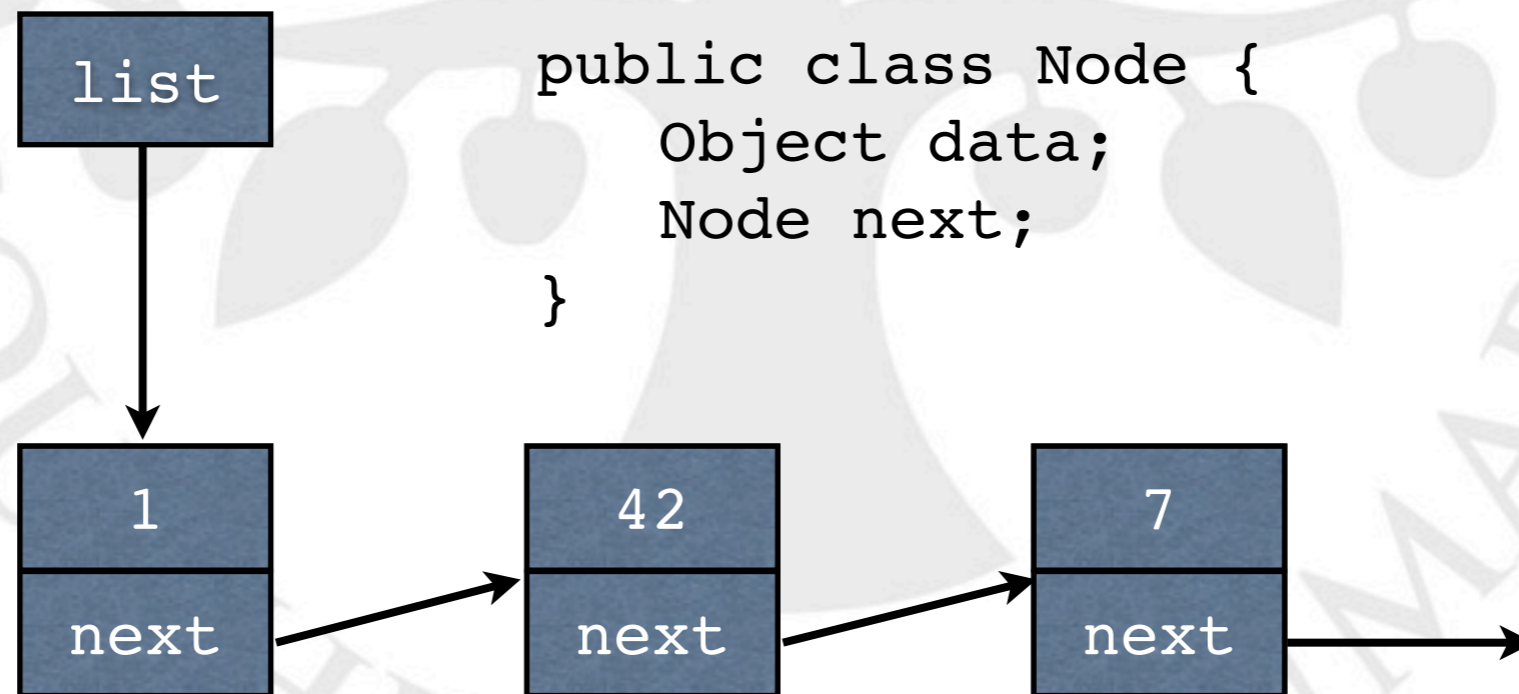
# Lister

- Enkelt-hægtet liste
  - Knuderne kan ligge hvor som helst i hukommelsen
  - Det sidste element peger på `null` (ingenting)



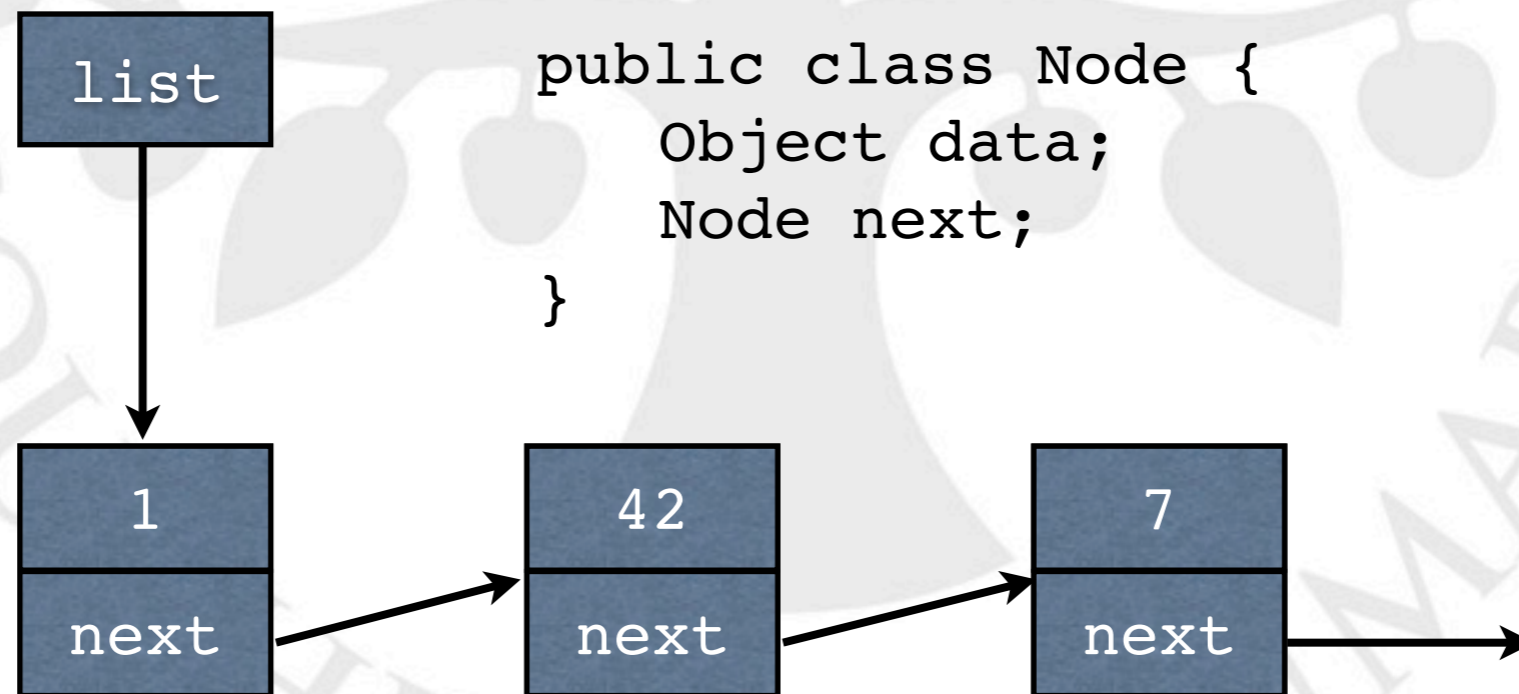
# Lister

- Enkelt-hægtet liste
  - Knuderne kan ligge hvor som helst i hukommelsen
  - Det sidste element peger på `null` (ingenting)
  - Bemærk at definitionen er rekursiv!

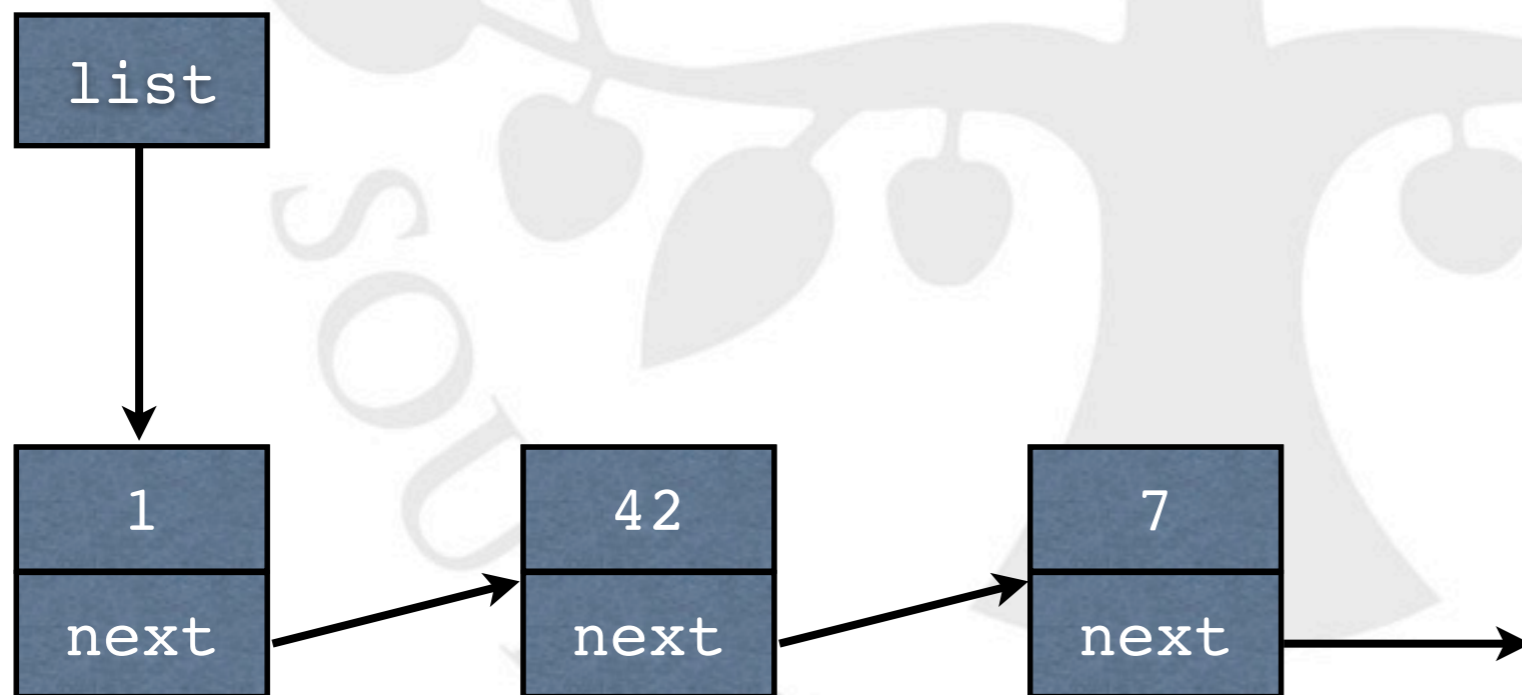


# Lister

- Enkelt-hægtet liste
  - Knuderne kan ligge hvor som helst i hukommelsen
  - Det sidste element peger på `null` (ingenting)
  - Bemærk at definitionen er rekursiv!
    - Meget lig definitionen af “liste af tal” fra sidst

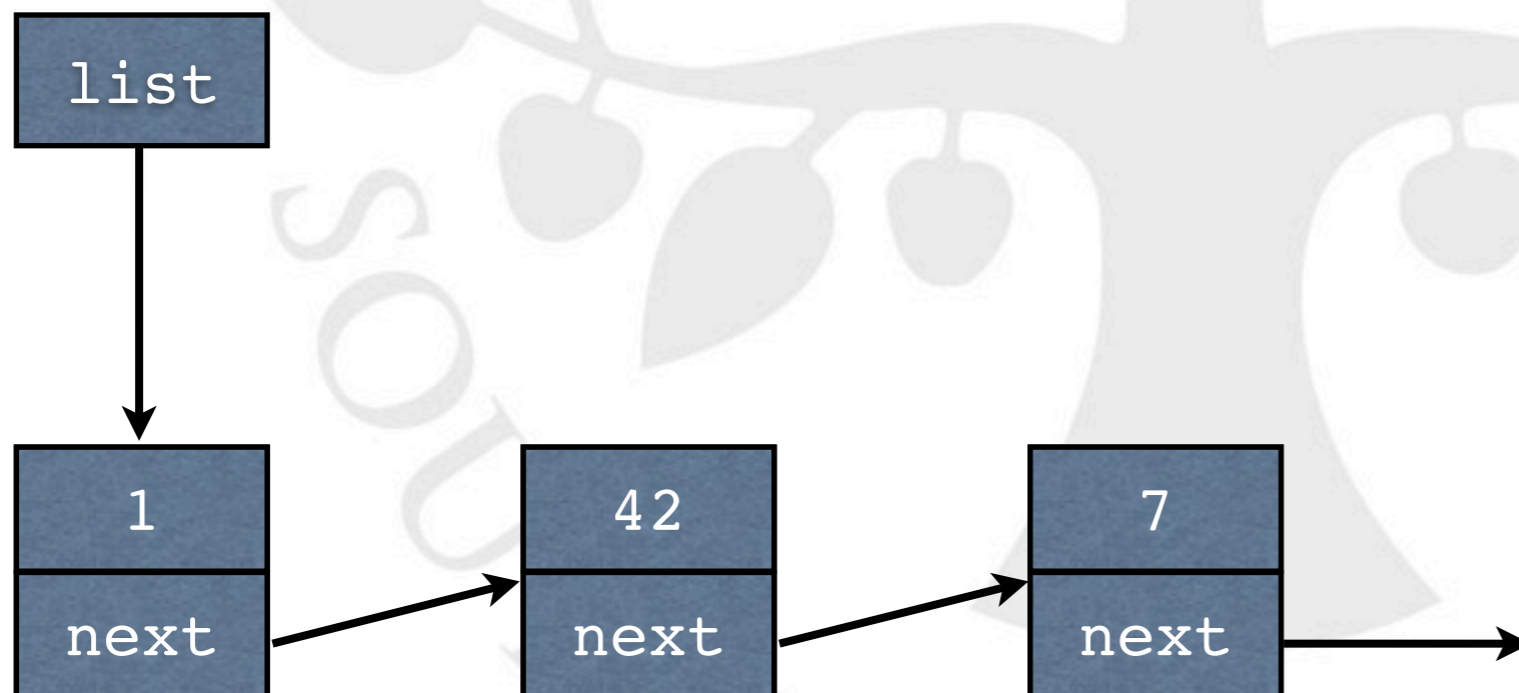


# Lister



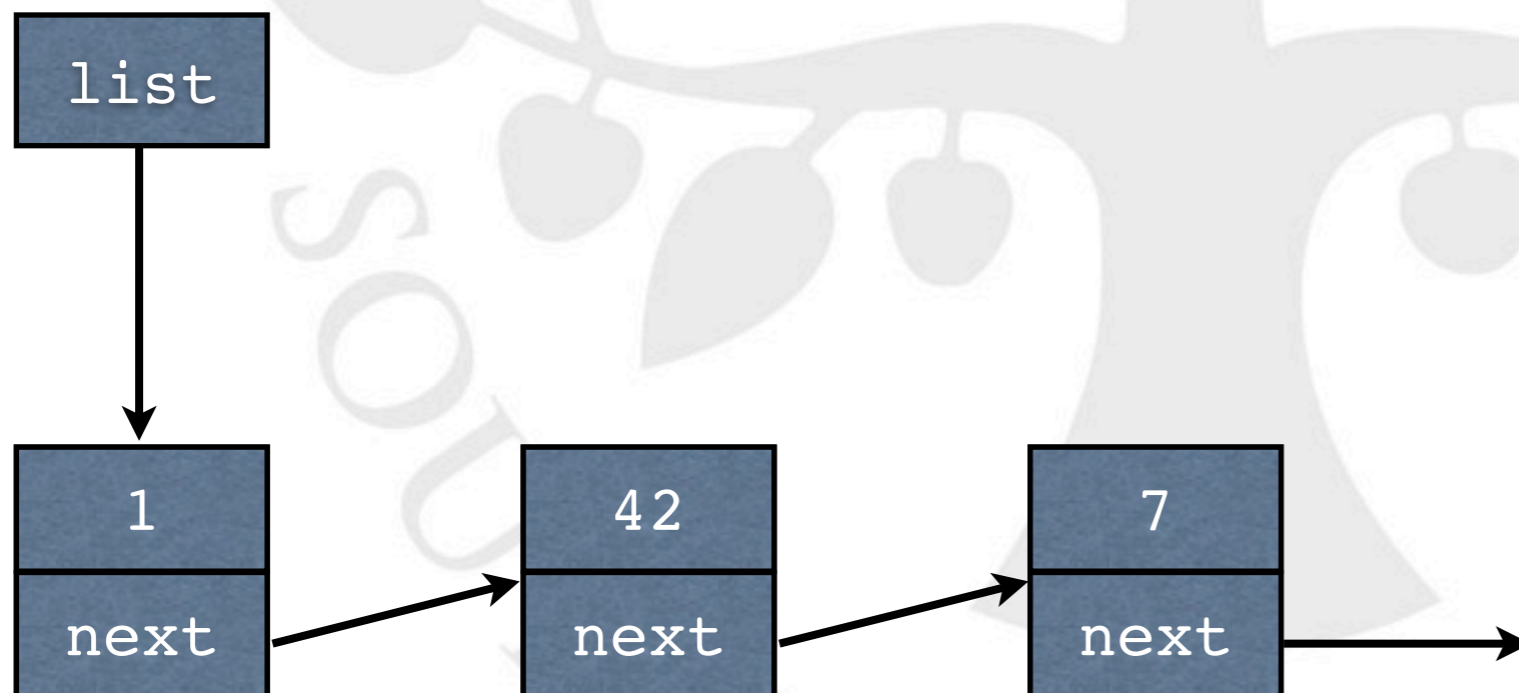
# Lister

- Indsættelse i enkelt-hægtet liste



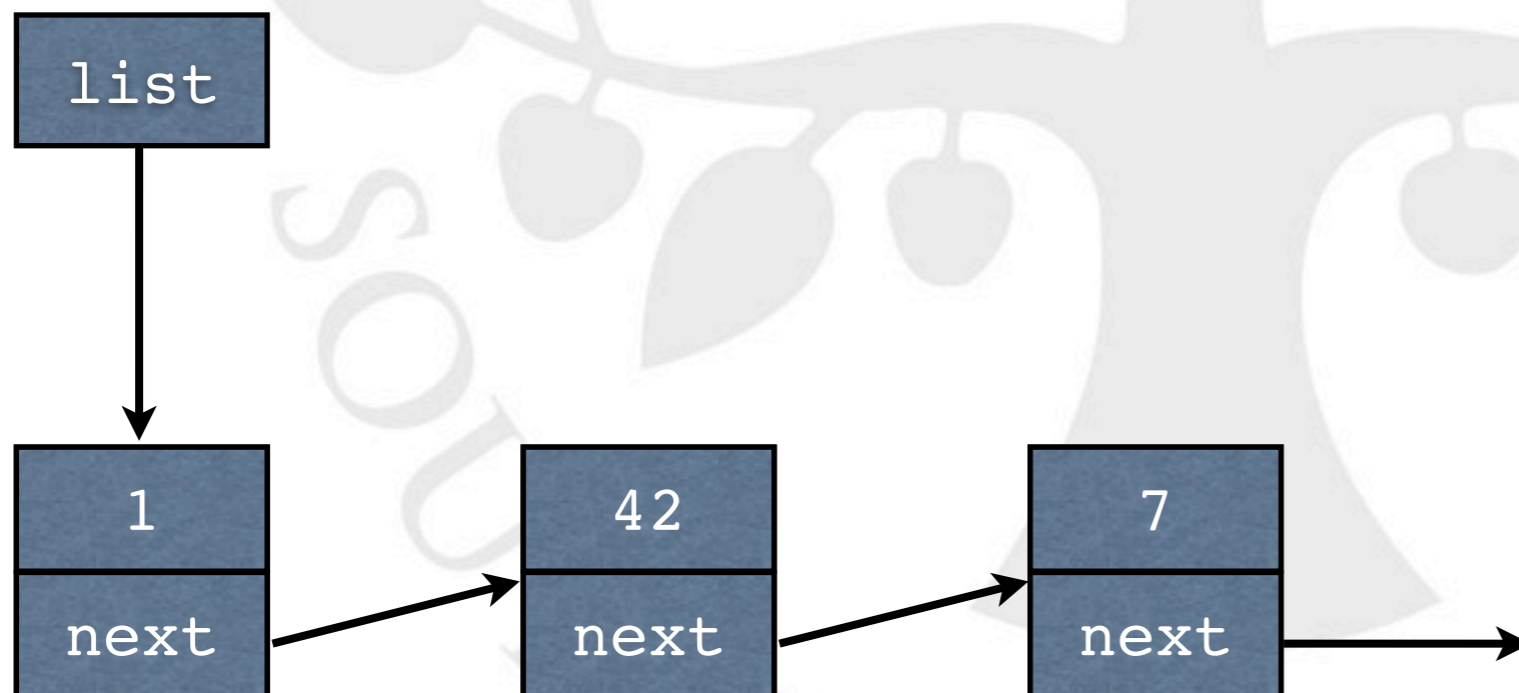
# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer



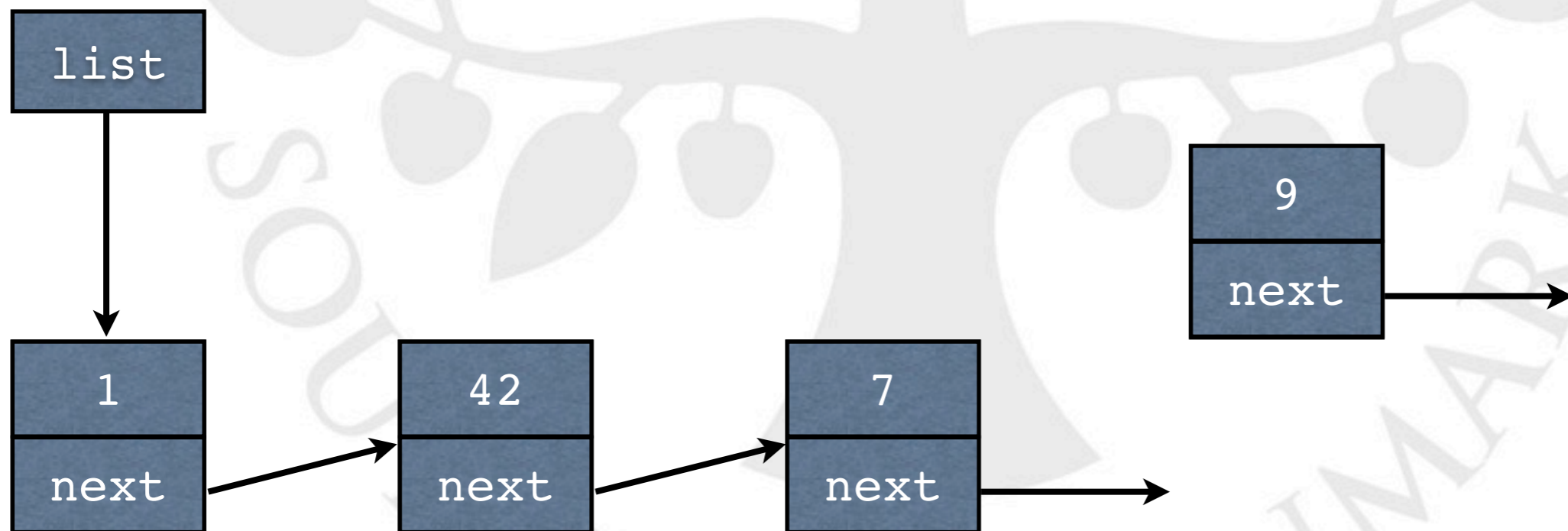
# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`



# Lister

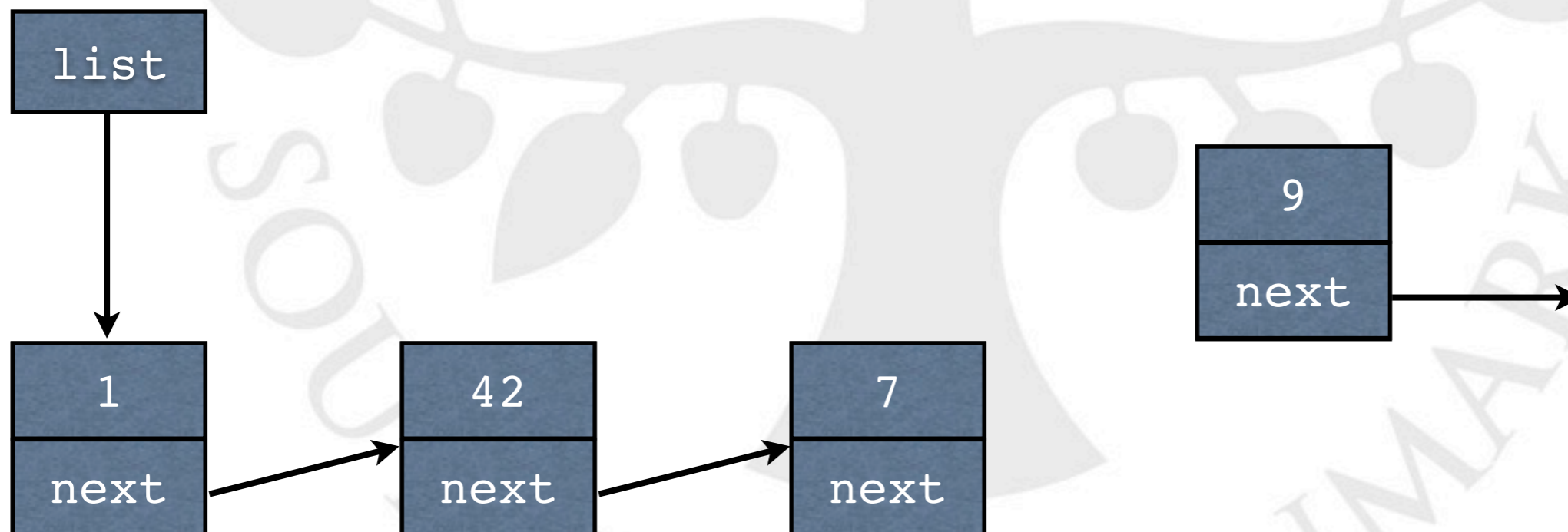
- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`





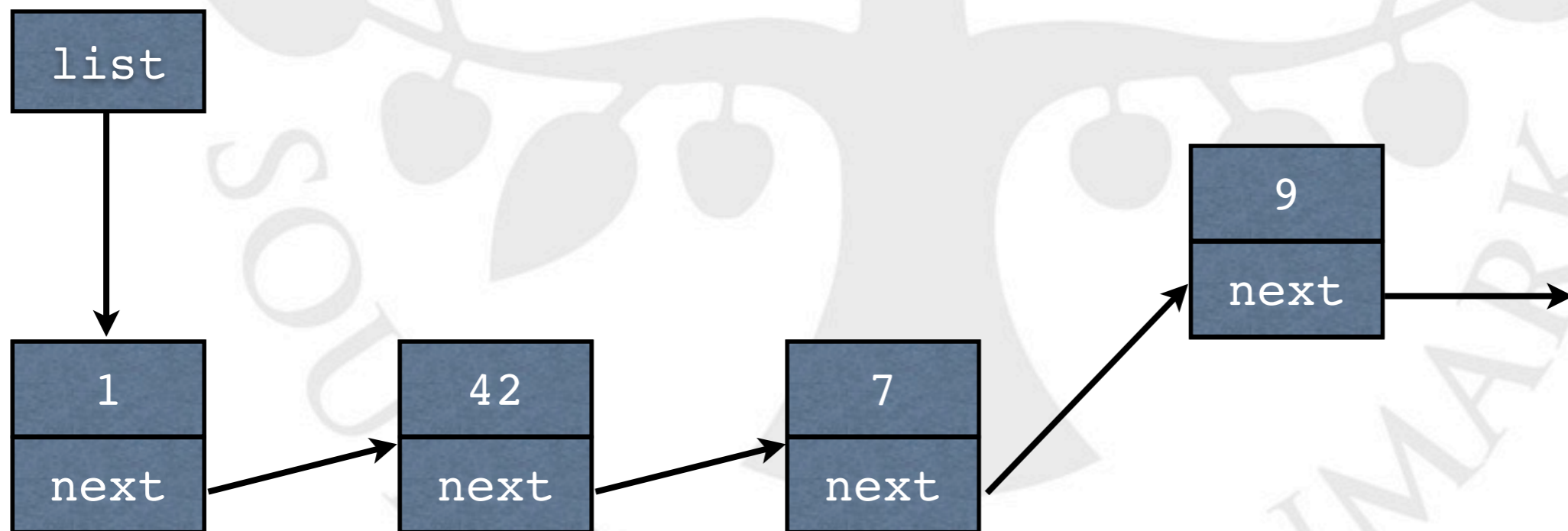
# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`



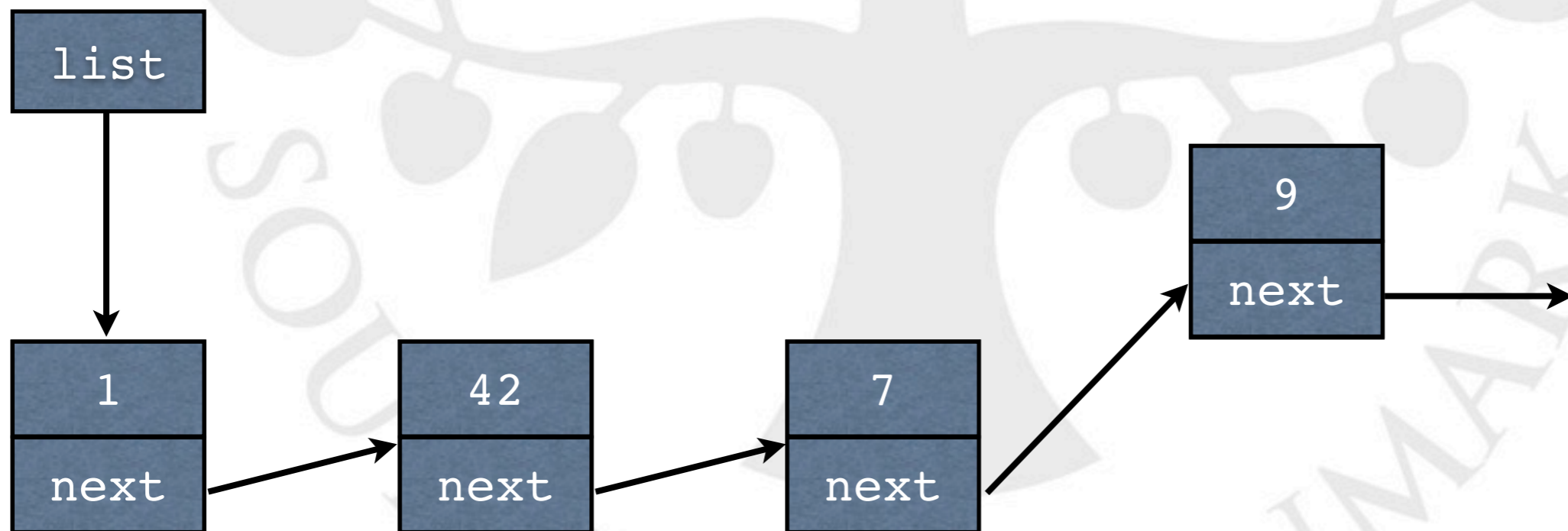
# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`



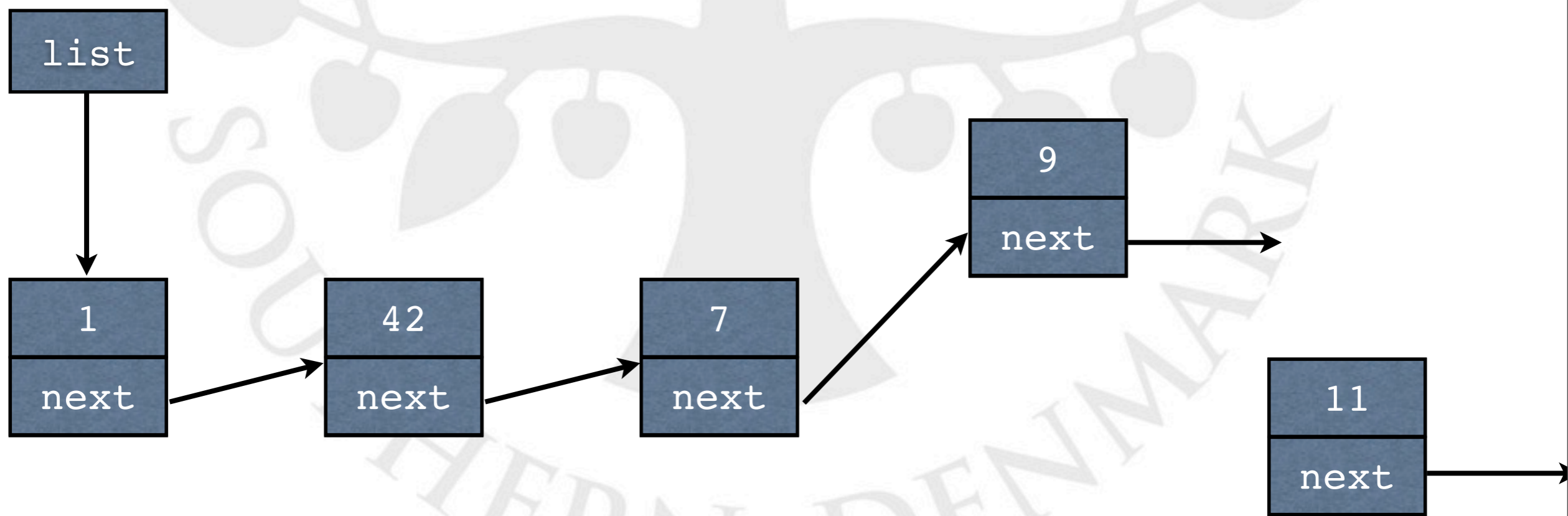
# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`
  - `add(11)`



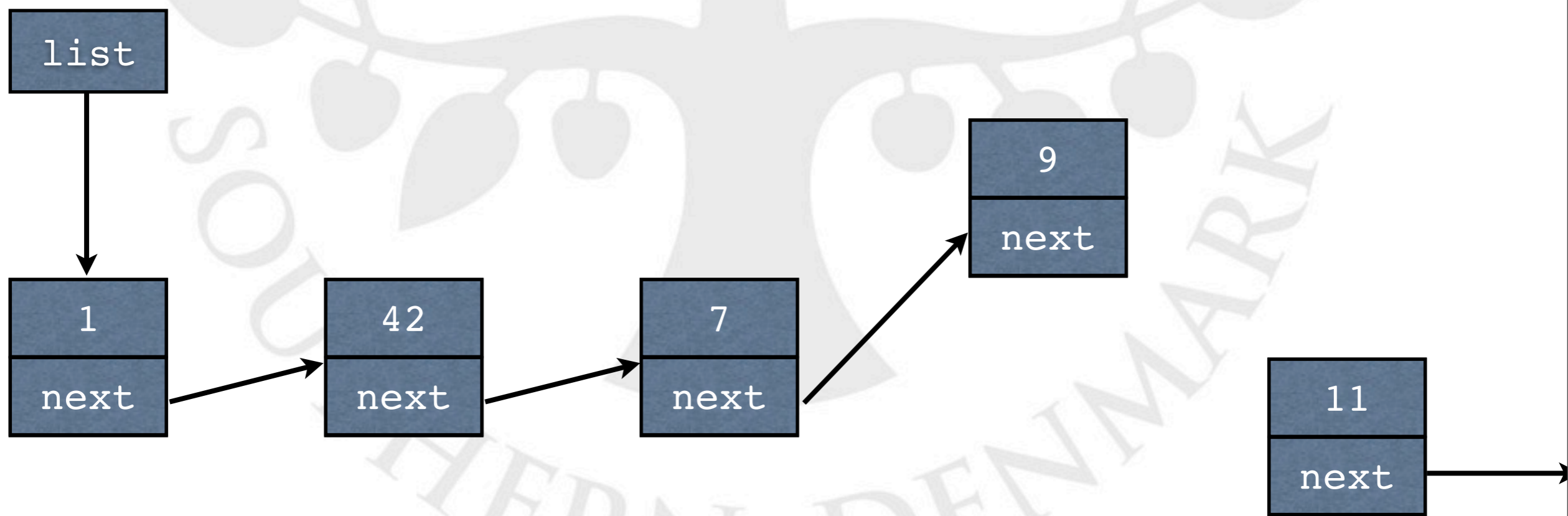
# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`
  - `add(11)`



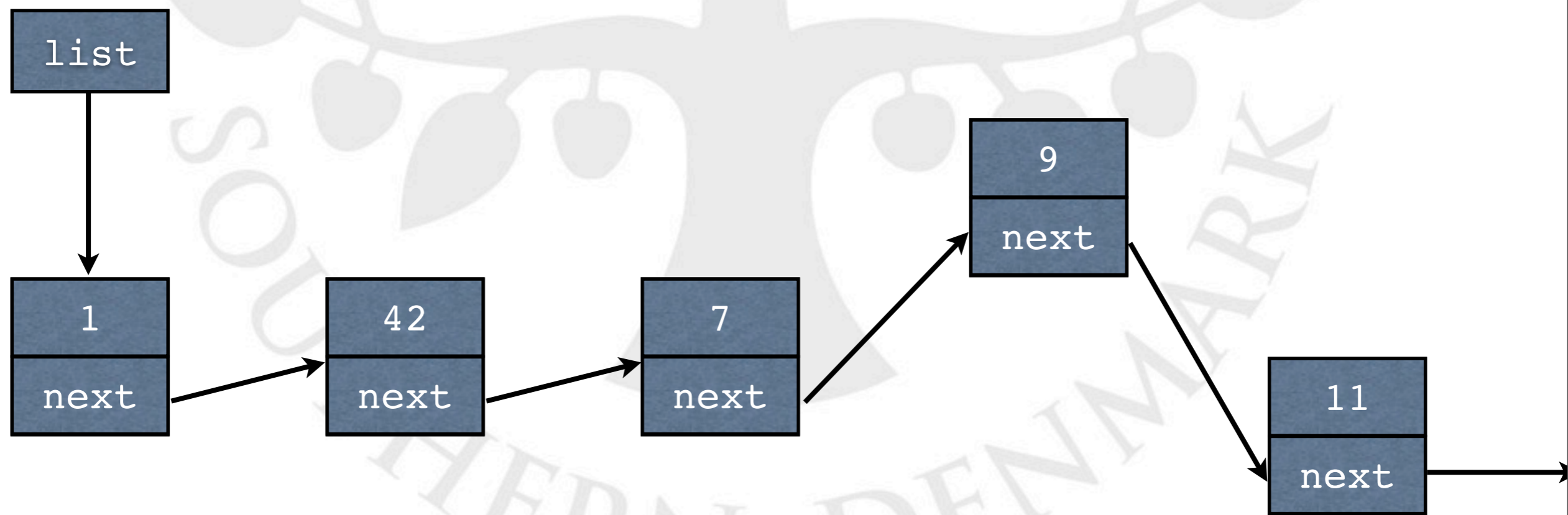
# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`
  - `add(11)`



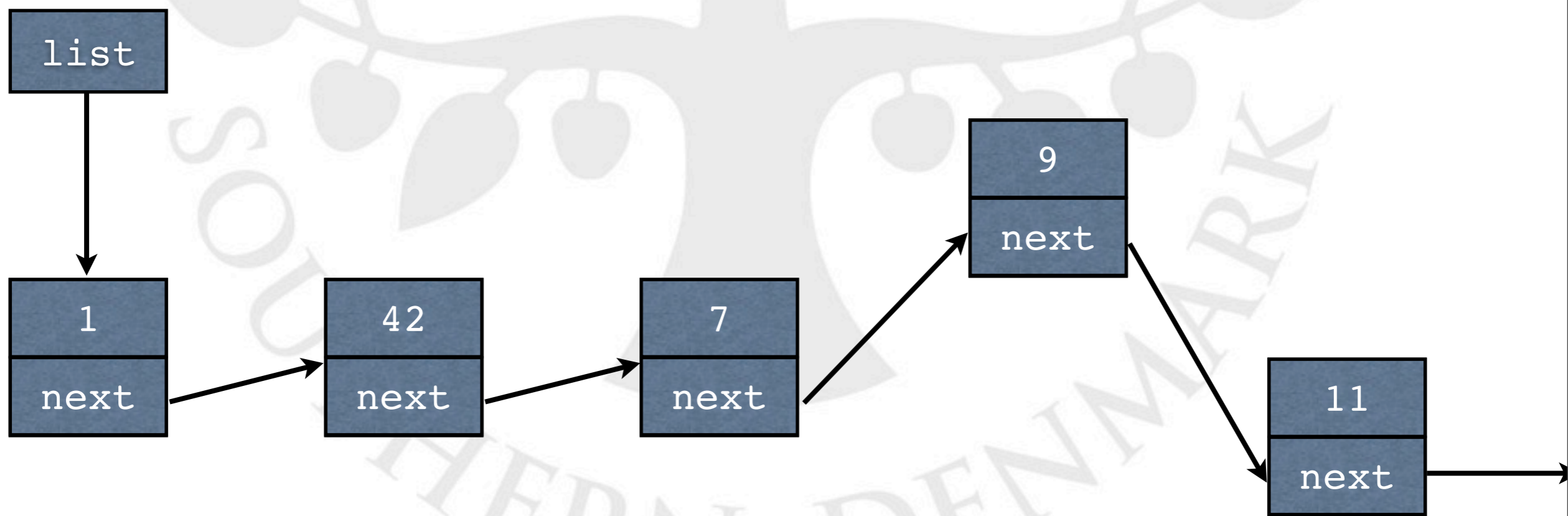
# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`
  - `add(11)`

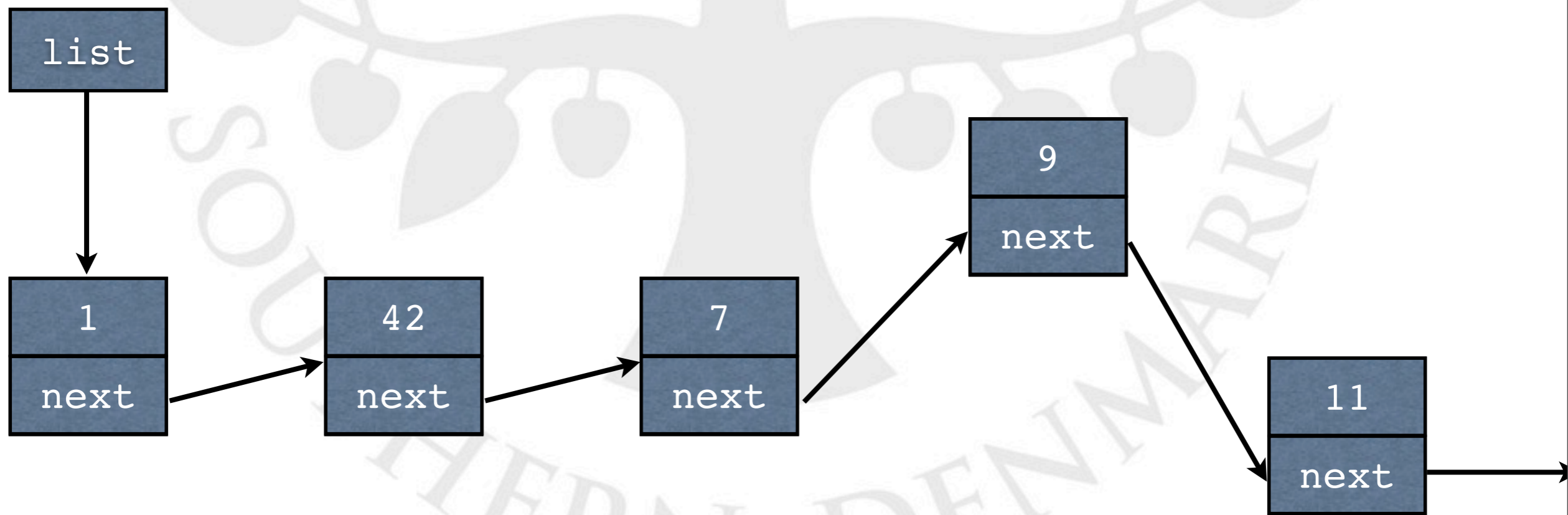


# Lister

- Indsættelse i enkelt-hægtet liste
- Sekvens af operationer
  - `add(9)`
  - `add(11)`
- Kan også indsætte midt i listen (overvej hvordan)



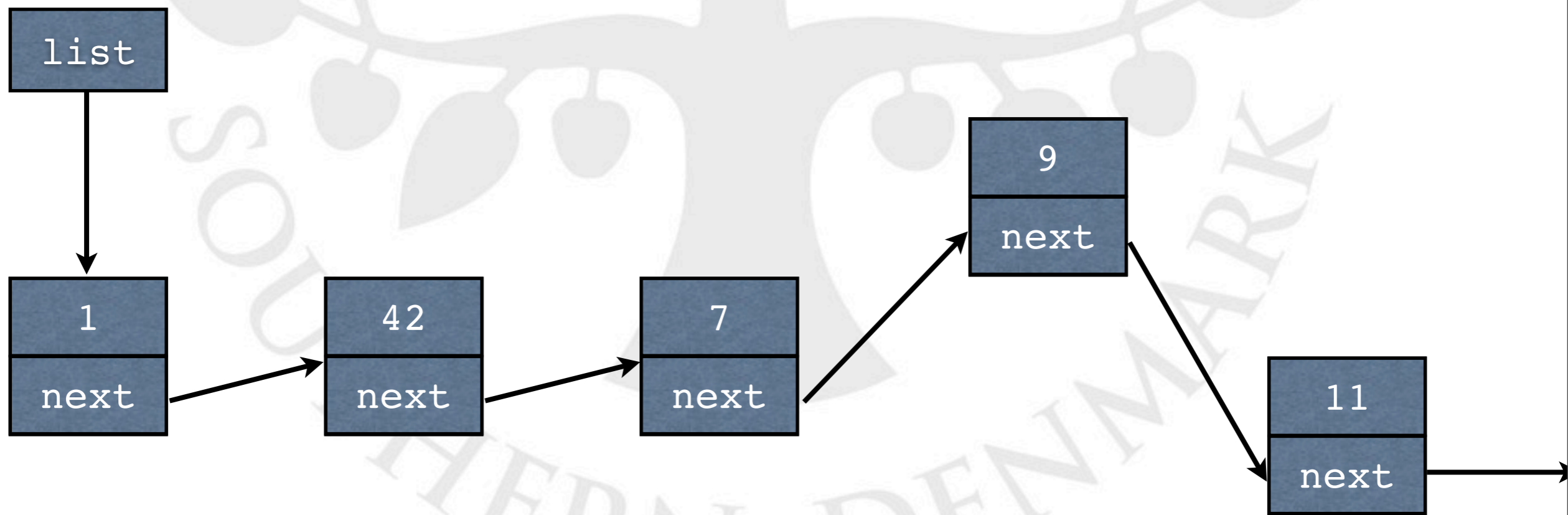
# Lister





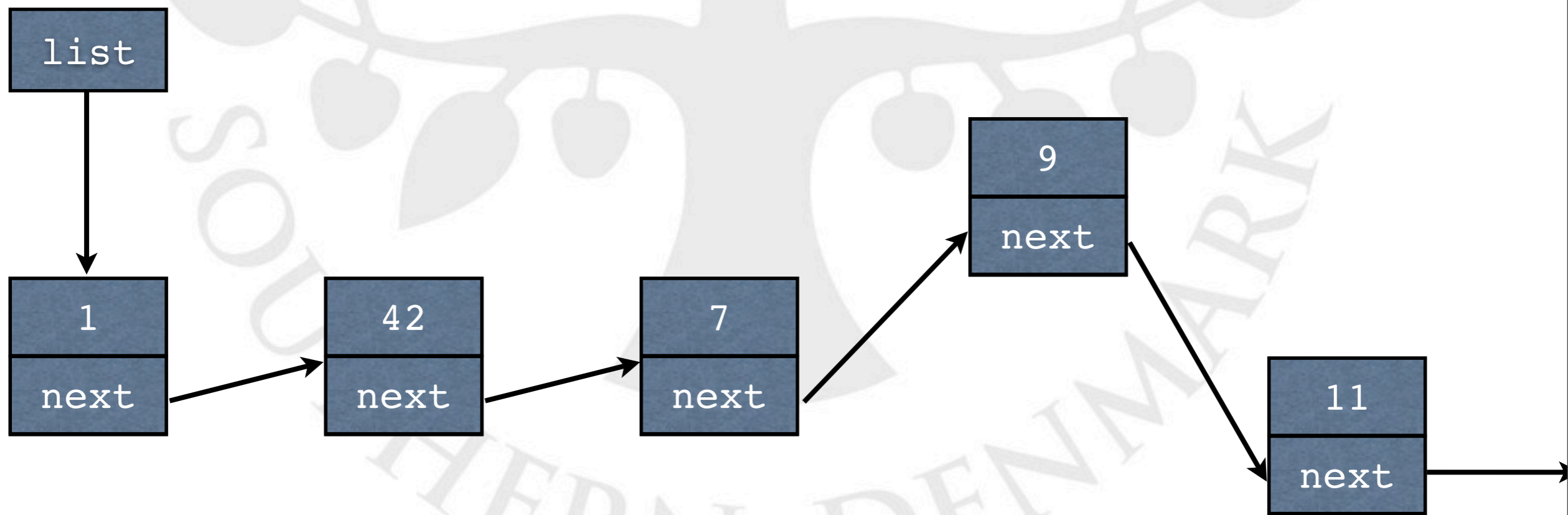
# Lister

- Sletning i enkelt-hægtet liste



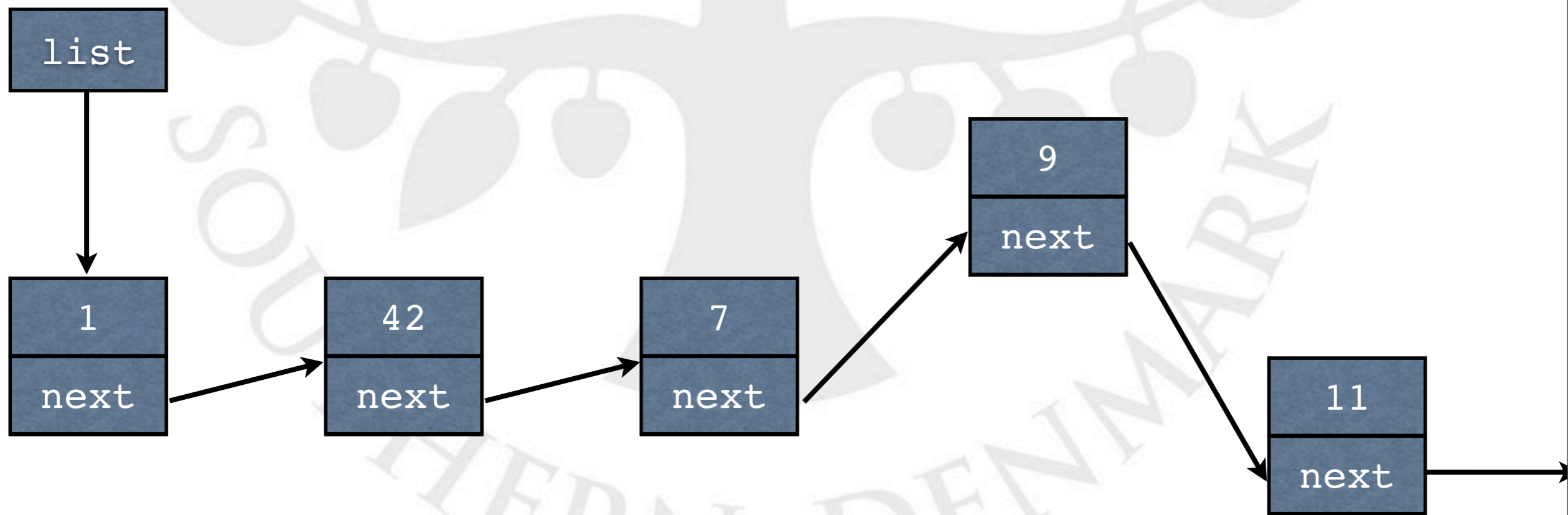
# Lister

- Sletning i enkelt-hægtet liste
- Sekvens af operationer



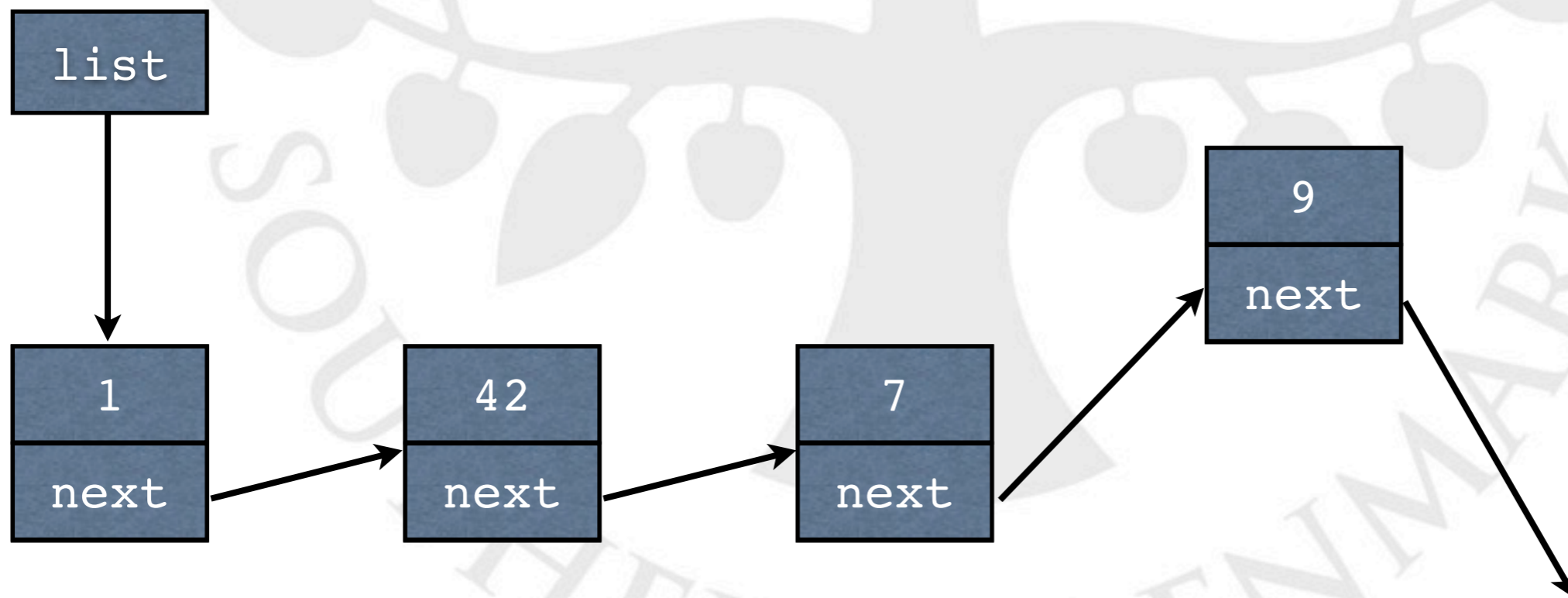
# Lister

- Sletning i enkelt-hægtet liste
- Sekvens af operationer
  - `remove(11)`



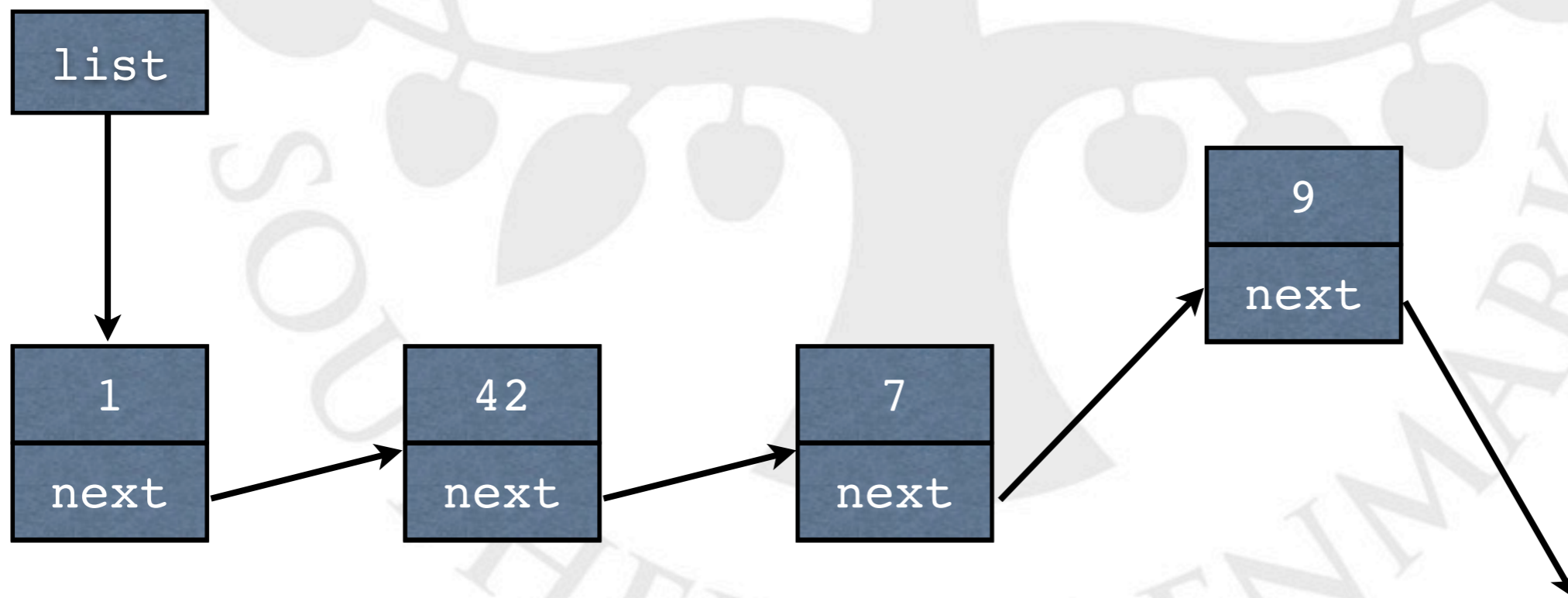
# Lister

- Sletning i enkelt-hægtet liste
- Sekvens af operationer
  - `remove(11)`



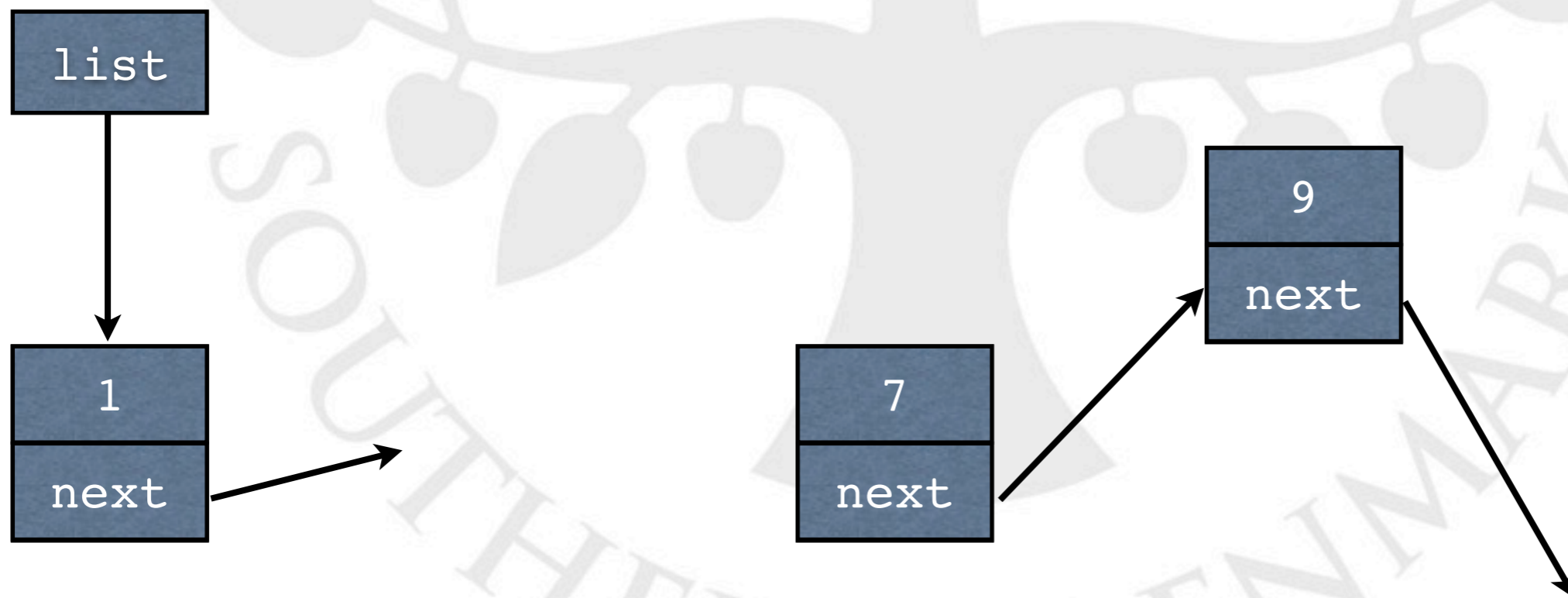
# Lister

- Sletning i enkelt-hægtet liste
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`



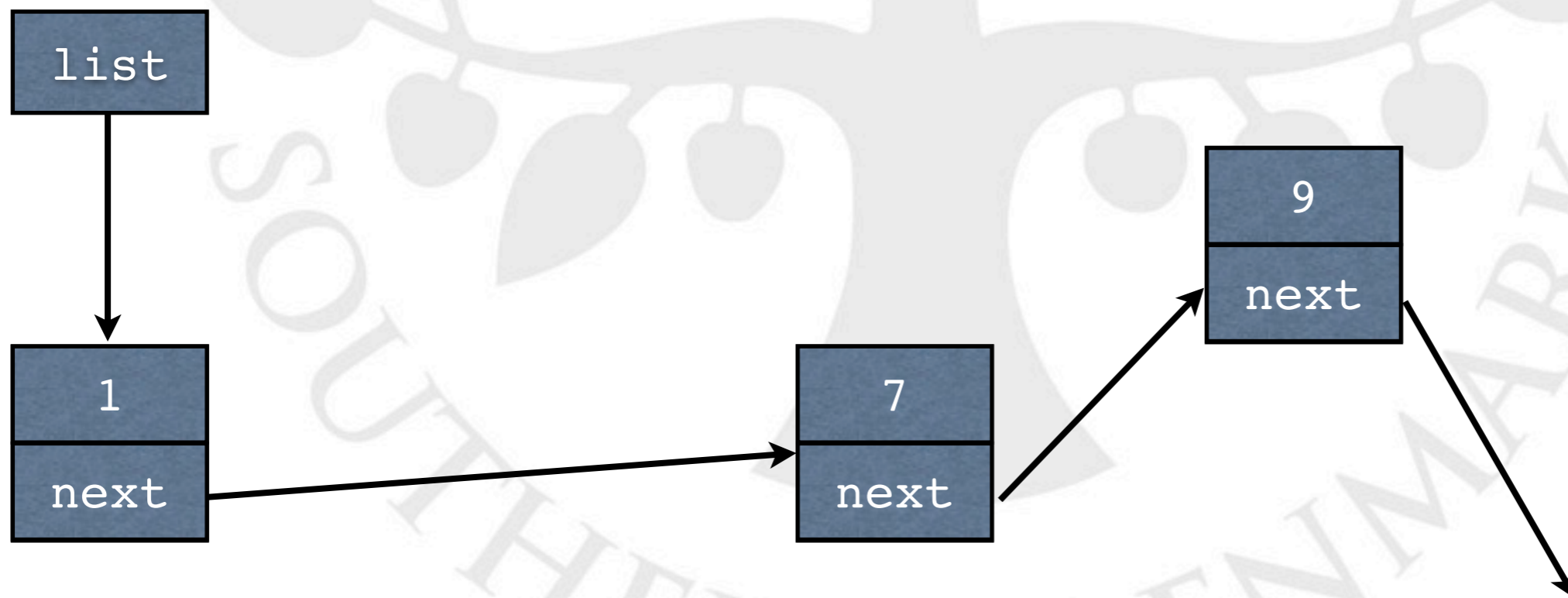
# Lister

- Sletning i enkelt-hægtet liste
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`

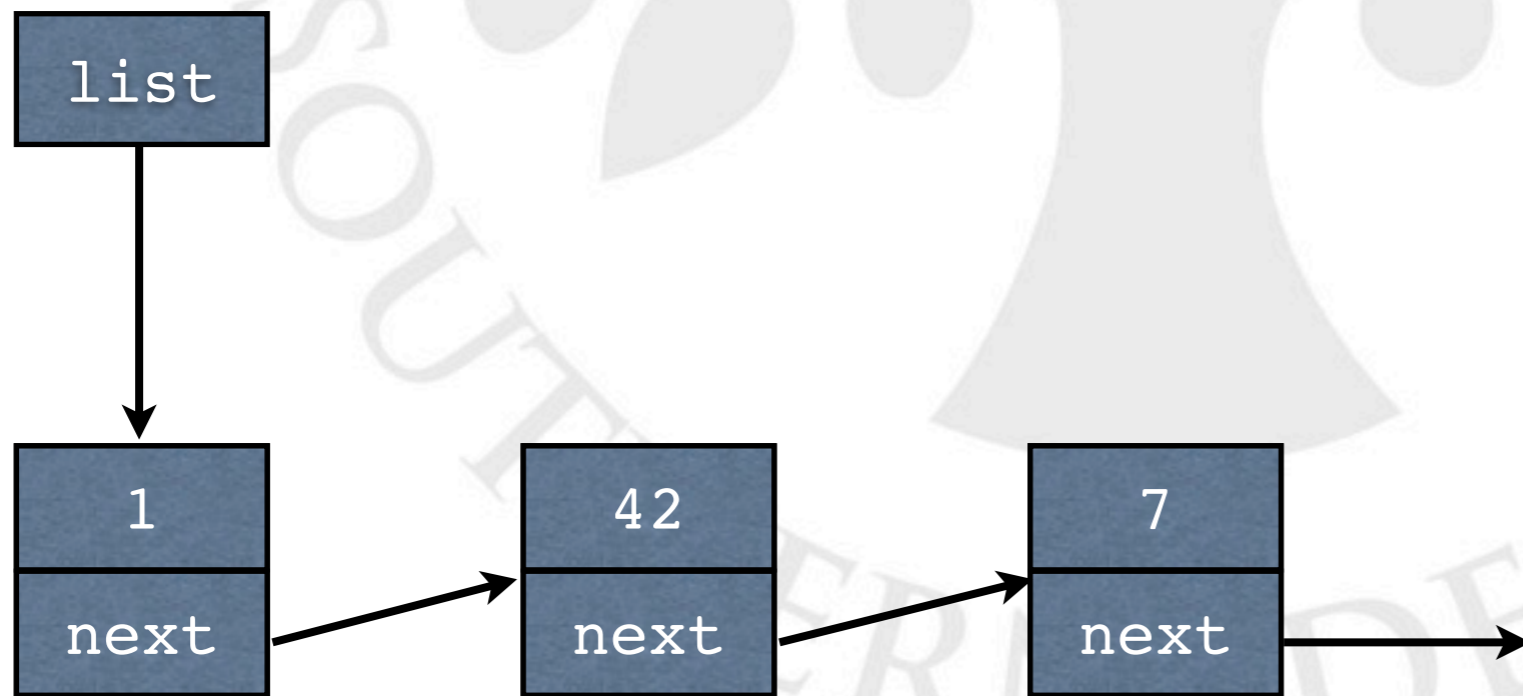


# Lister

- Sletning i enkelt-hægtet liste
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`



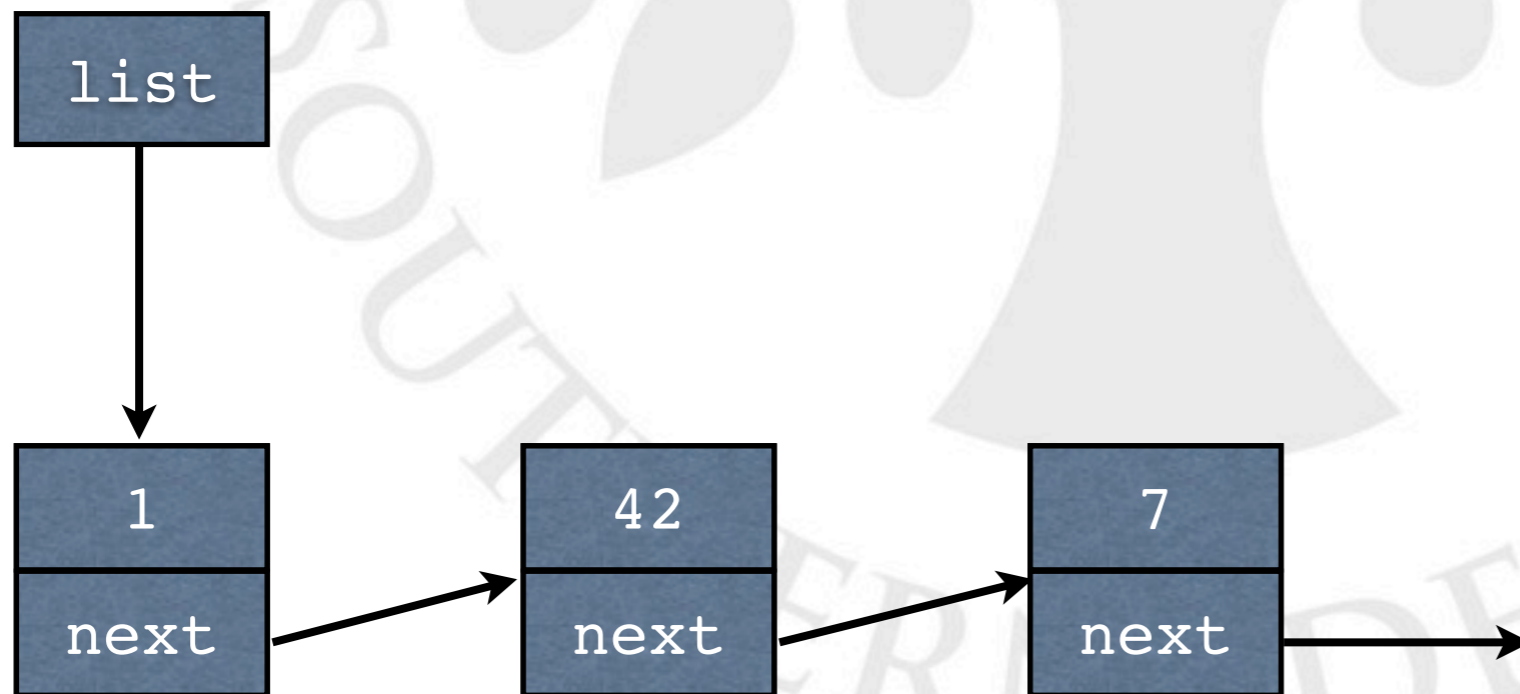
# Lister





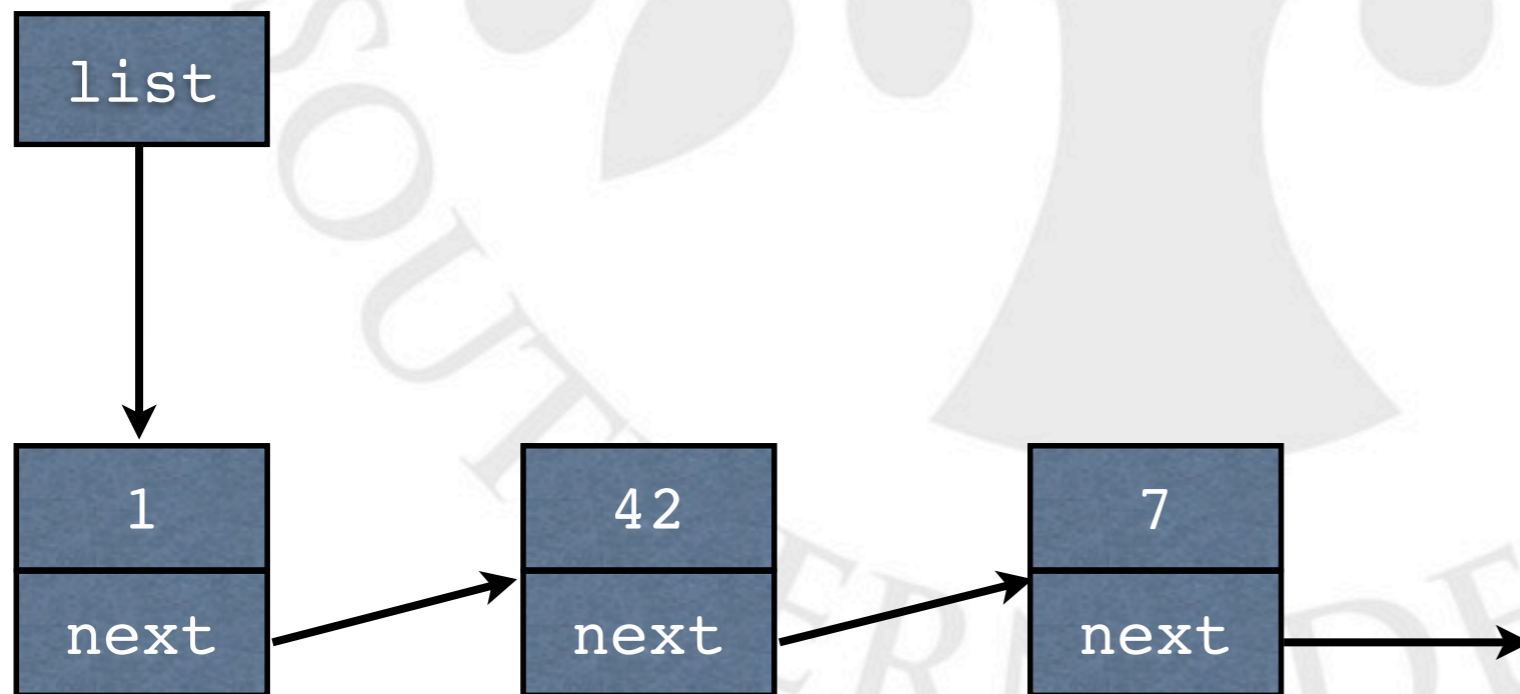
# Lister

- Enkelt-hægtet liste



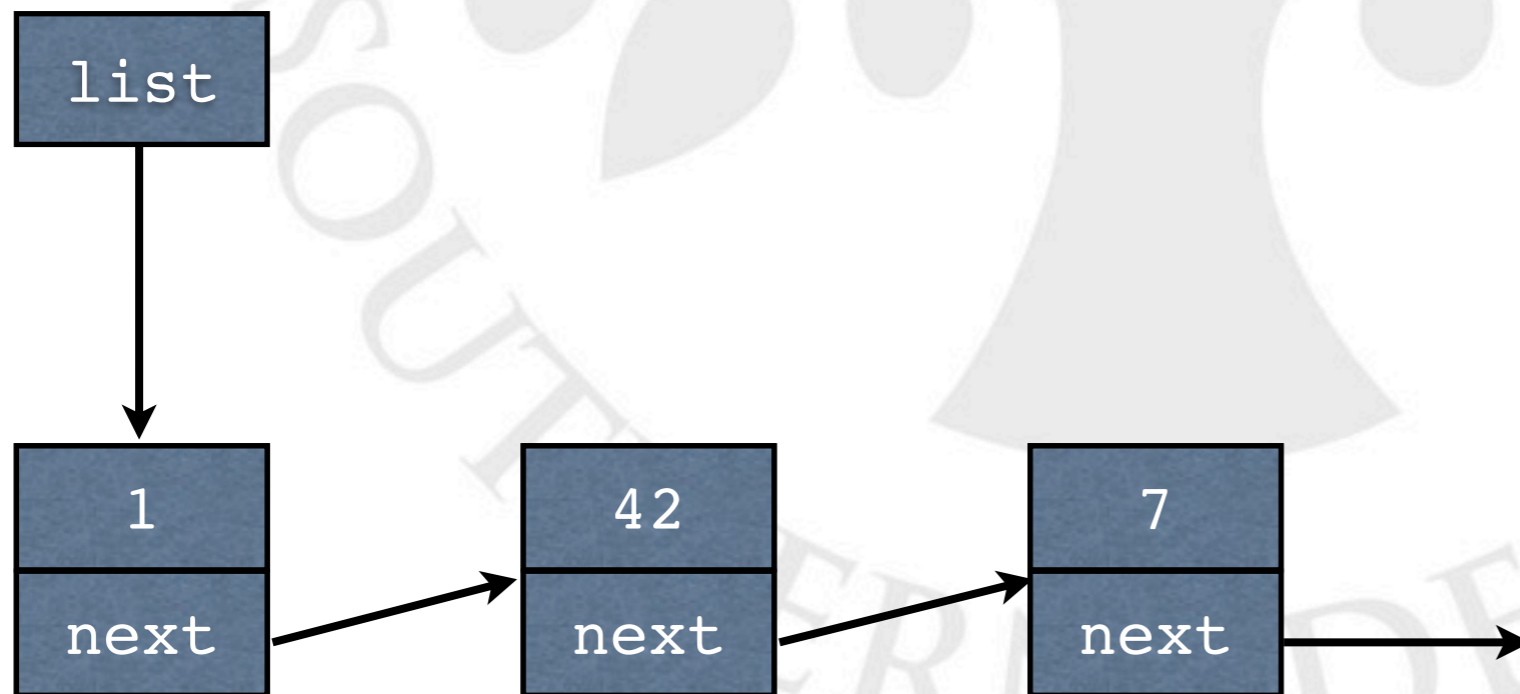
# Lister

- Enkelt-hægtet liste
  - Alle operationer skal løbe listen igennem...?



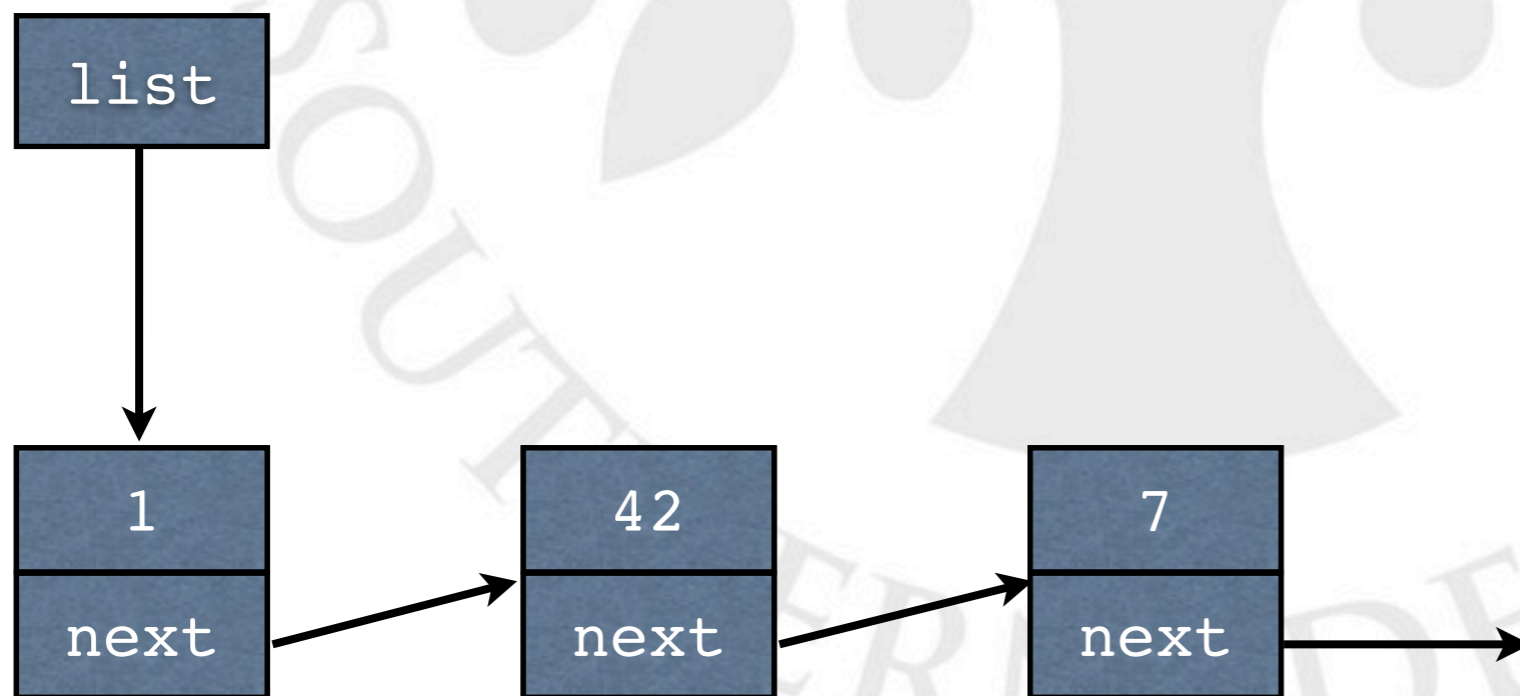
# Lister

- Enkelt-hægtet liste
  - Alle operationer skal løbe listen igennem...?
    - På nær indsættelse og sletning i begyndelsen



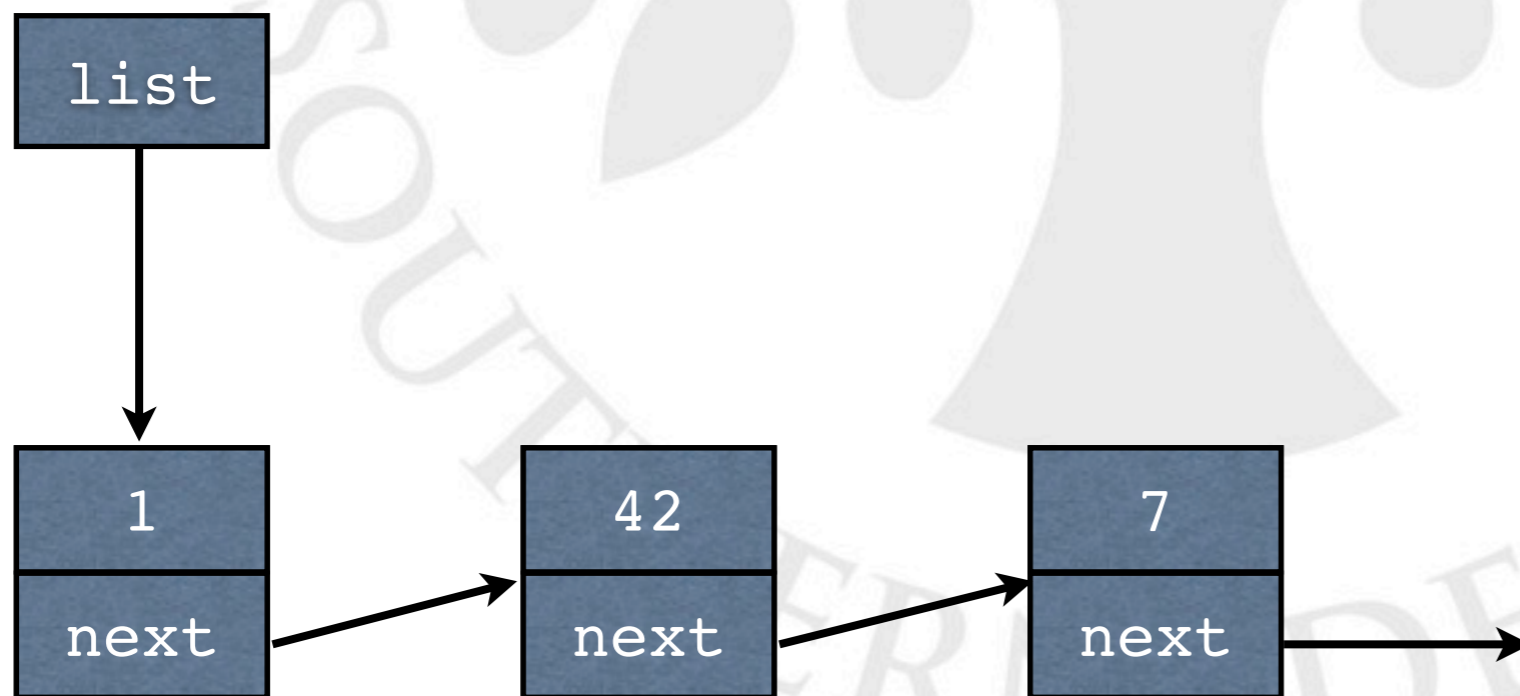
# Lister

- Enkelt-hægtet liste
  - Alle operationer skal løbe listen igennem...?
    - På nær indsættelse og sletning i begyndelsen
  - Kan kun løbe listen igennem i en retning



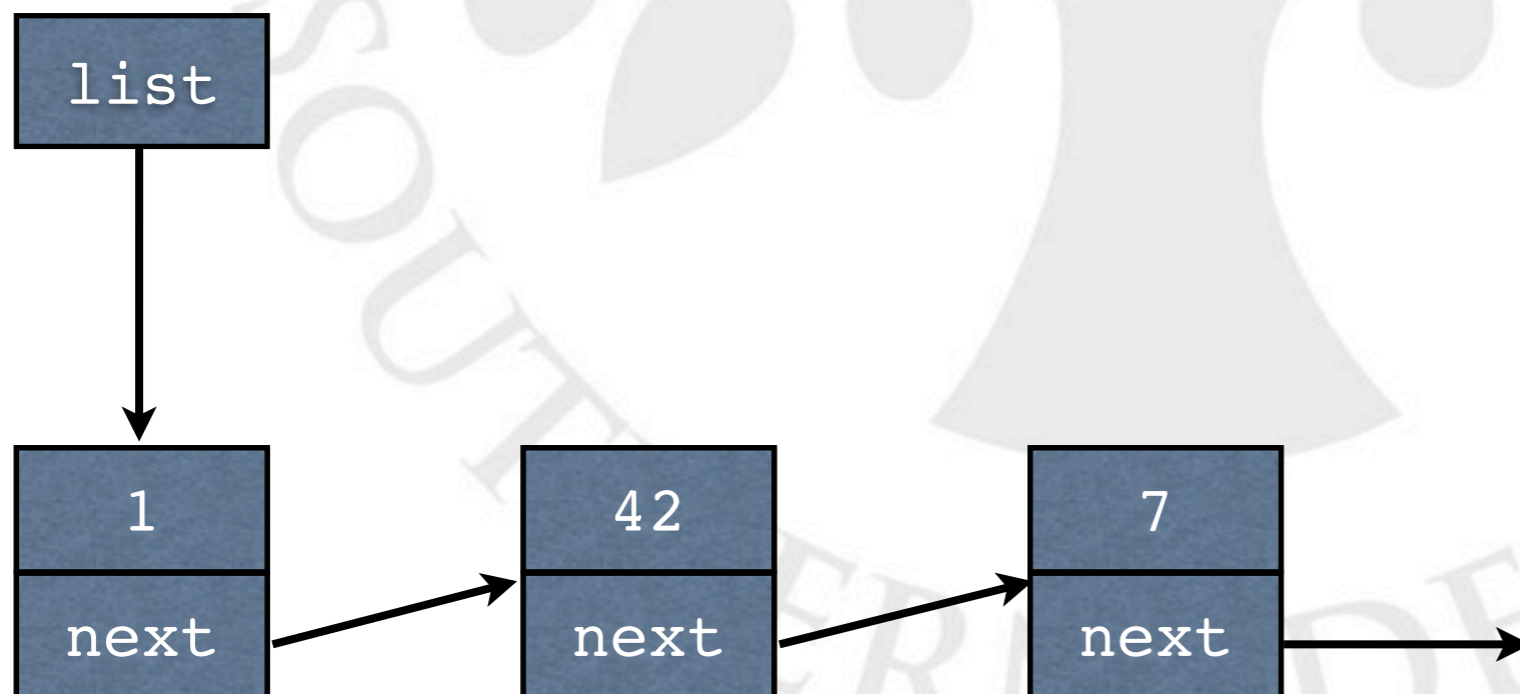
# Lister

- Enkelt-hægtet liste
  - Alle operationer skal løbe listen igennem...?
    - På nær indsættelse og sletning i begyndelsen
  - Kan kun løbe listen igennem i en retning
    - Man kan ikke afgøre hvilket element der ligger lige før et givet element

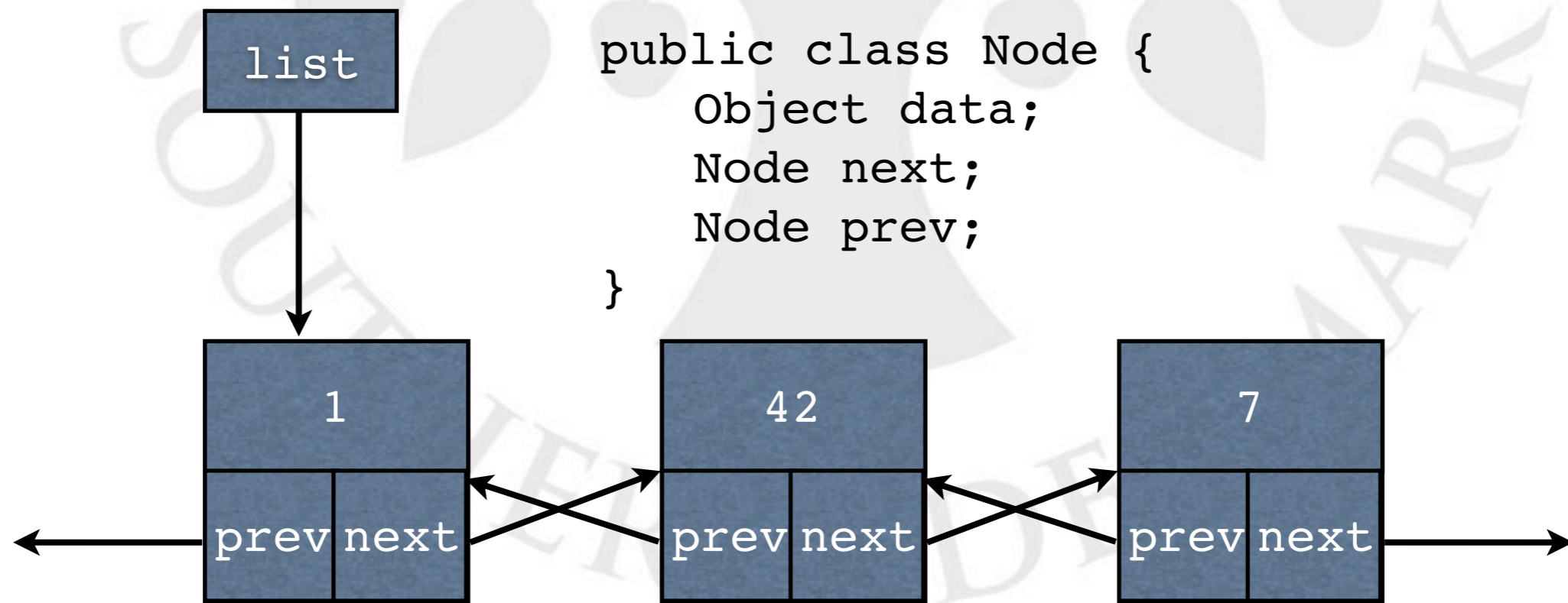


# Lister

- Enkelt-hægtet liste
  - Alle operationer skal løbe listen igennem...?
    - På nær indsættelse og sletning i begyndelsen
  - Kan kun løbe listen igennem i en retning
    - Man kan ikke afgøre hvilket element der ligger lige før et givet element
      - Ingen prev(e) operation

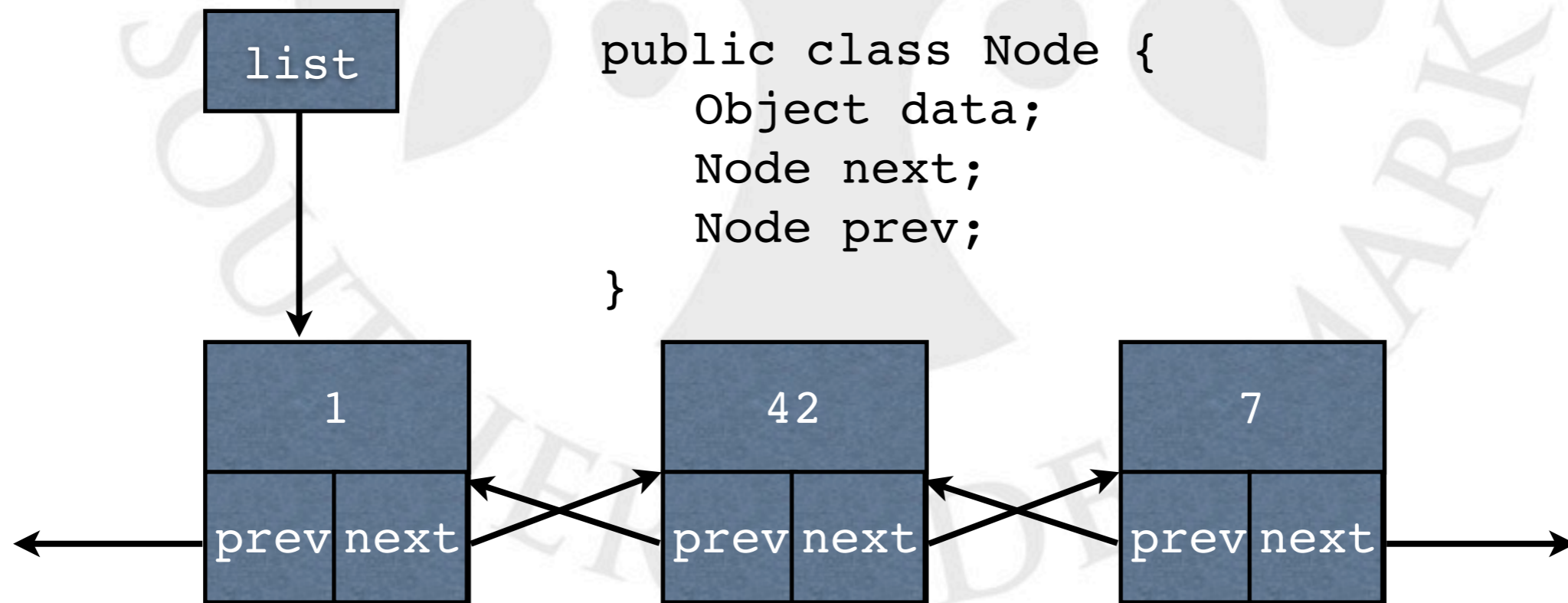


# Lister



# Lister

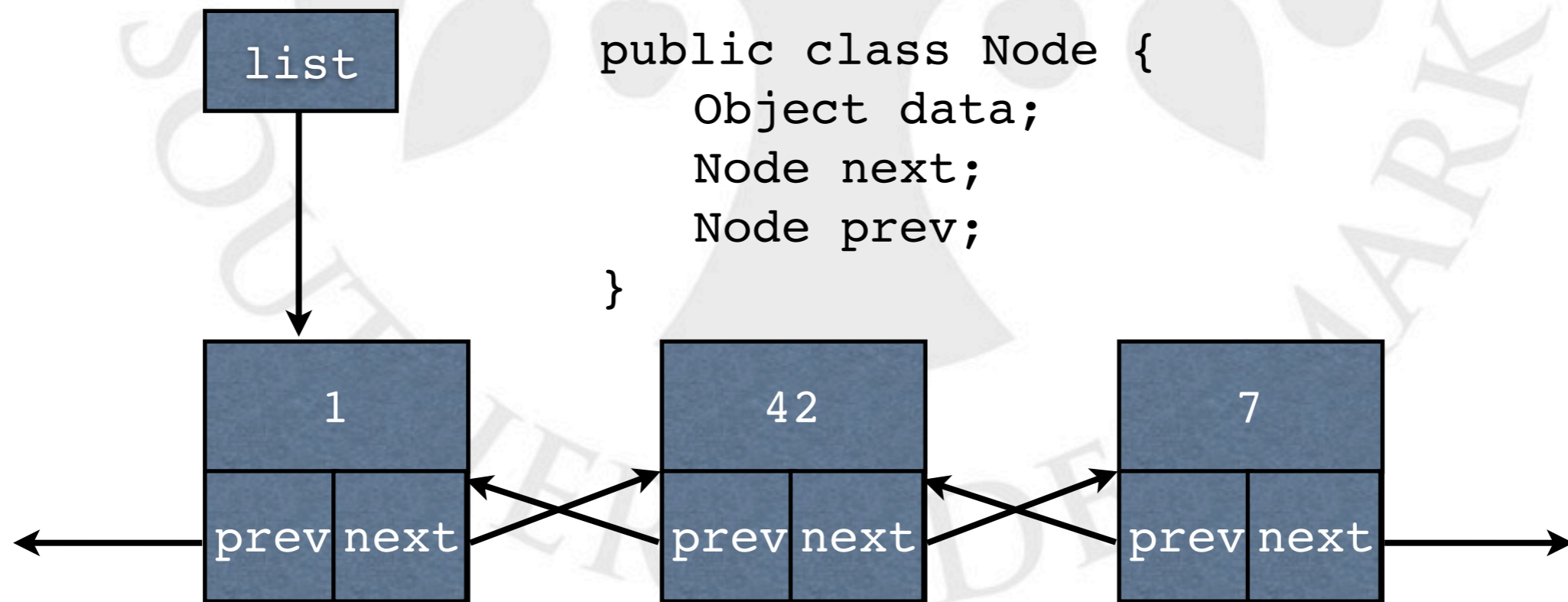
- Dobbelt-hægtet liste





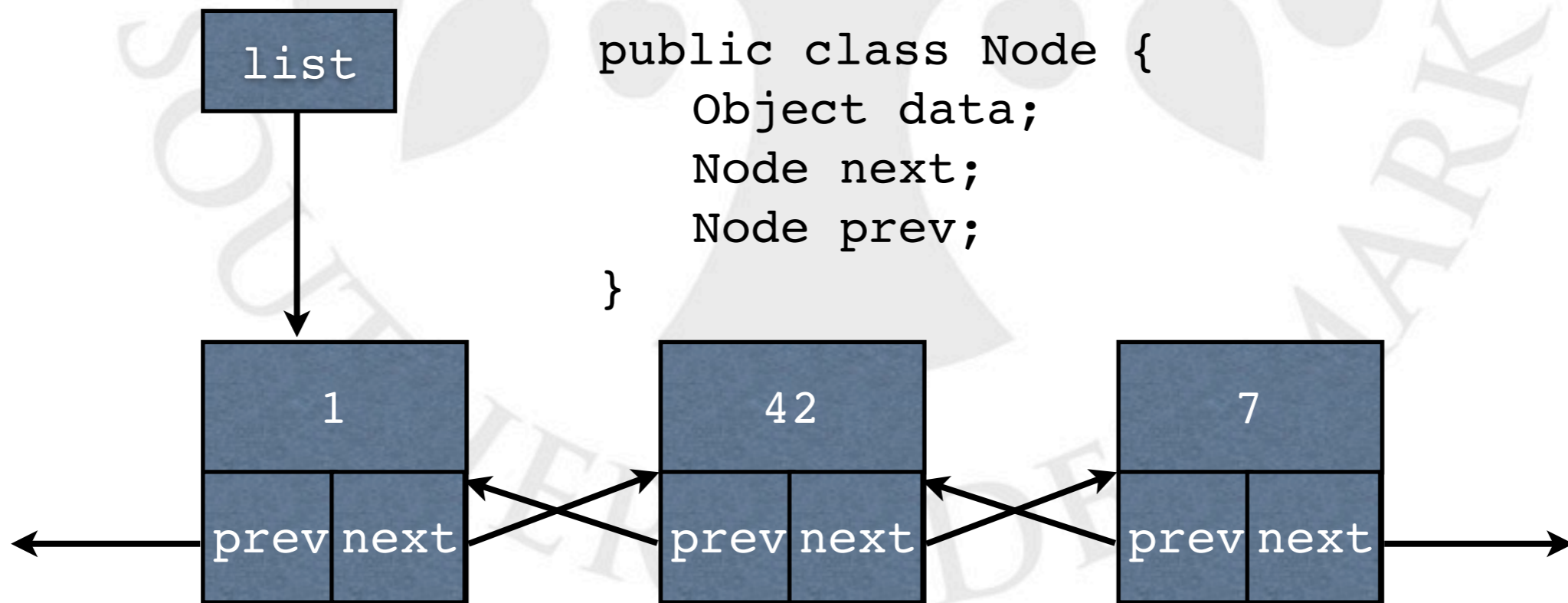
# Lister

- Dobbelt-hægtet liste
  - Ligesom enkelt-hægtet liste, men med et ekstra felt



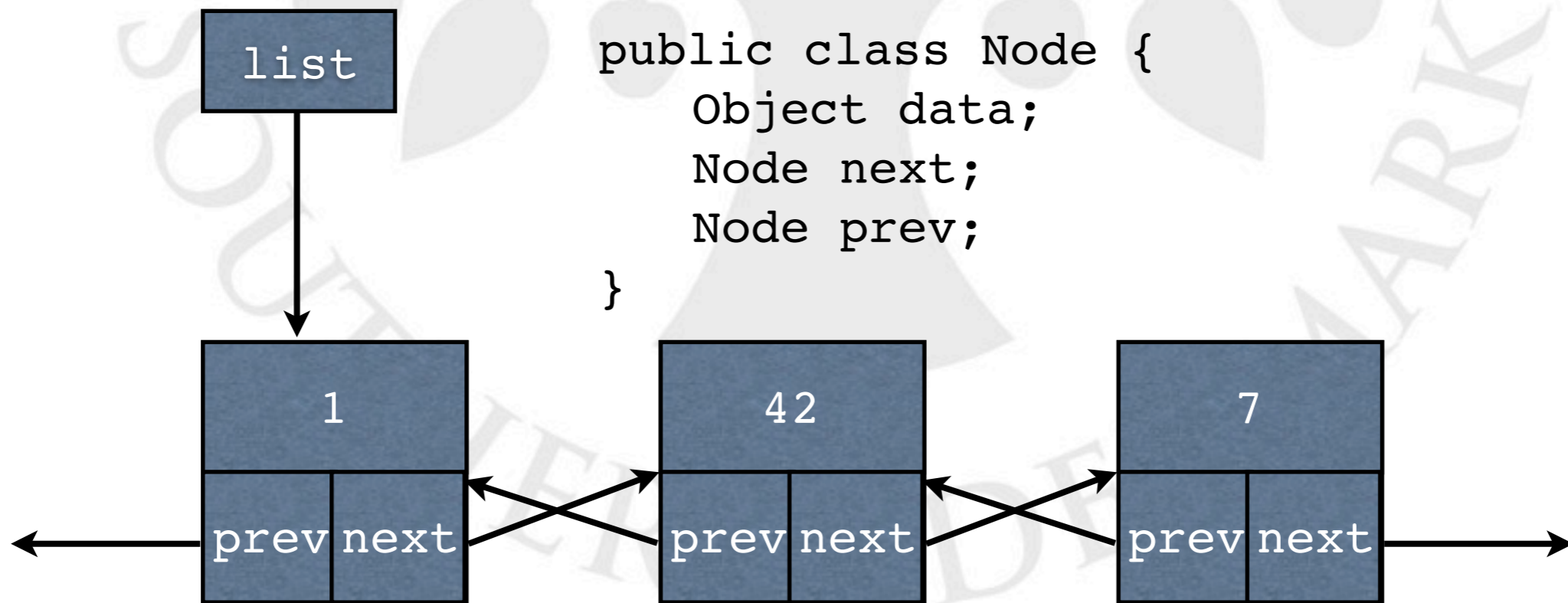
# Lister

- Dobbelt-hægtet liste
  - Ligesom enkelt-hægtet liste, men med et ekstra felt
  - Lige så effektiv som enkelt-hægtet liste



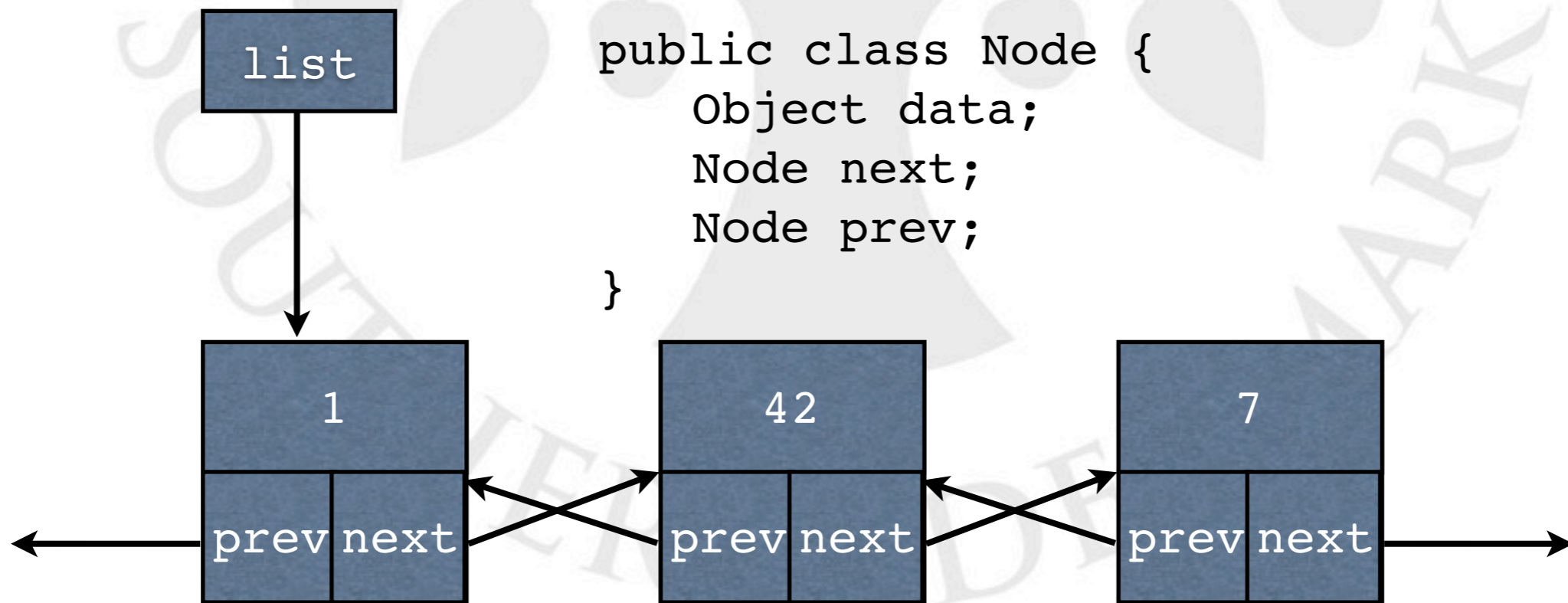
# Lister

- Dobbelt-hægtet liste
  - Ligesom enkelt-hægtet liste, men med et ekstra felt
  - Lige så effektiv som enkelt-hægtet liste
  - Kan nu løbe igennem begge veje



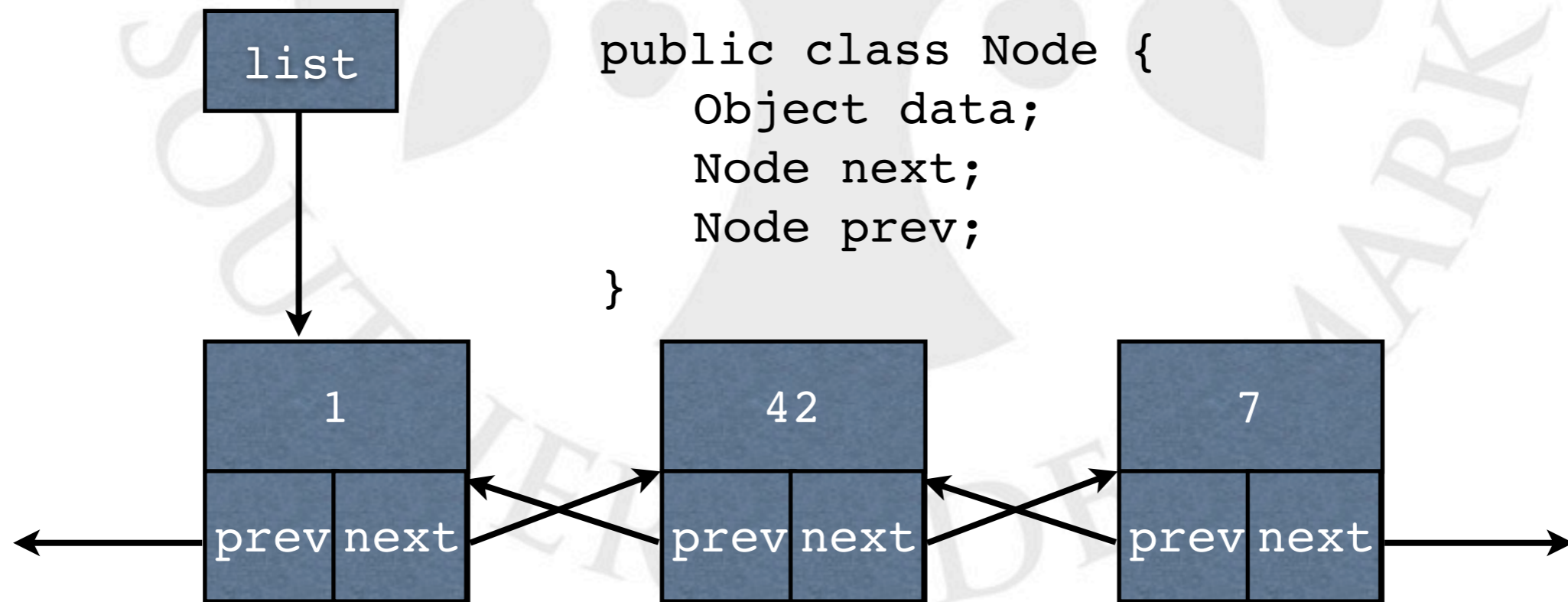
# Lister

- Dobbelt-hægtet liste
  - Ligesom enkelt-hægtet liste, men med et ekstra felt
  - Lige så effektiv som enkelt-hægtet liste
  - Kan nu løbe igennem begge veje
- `java.util.LinkedList`



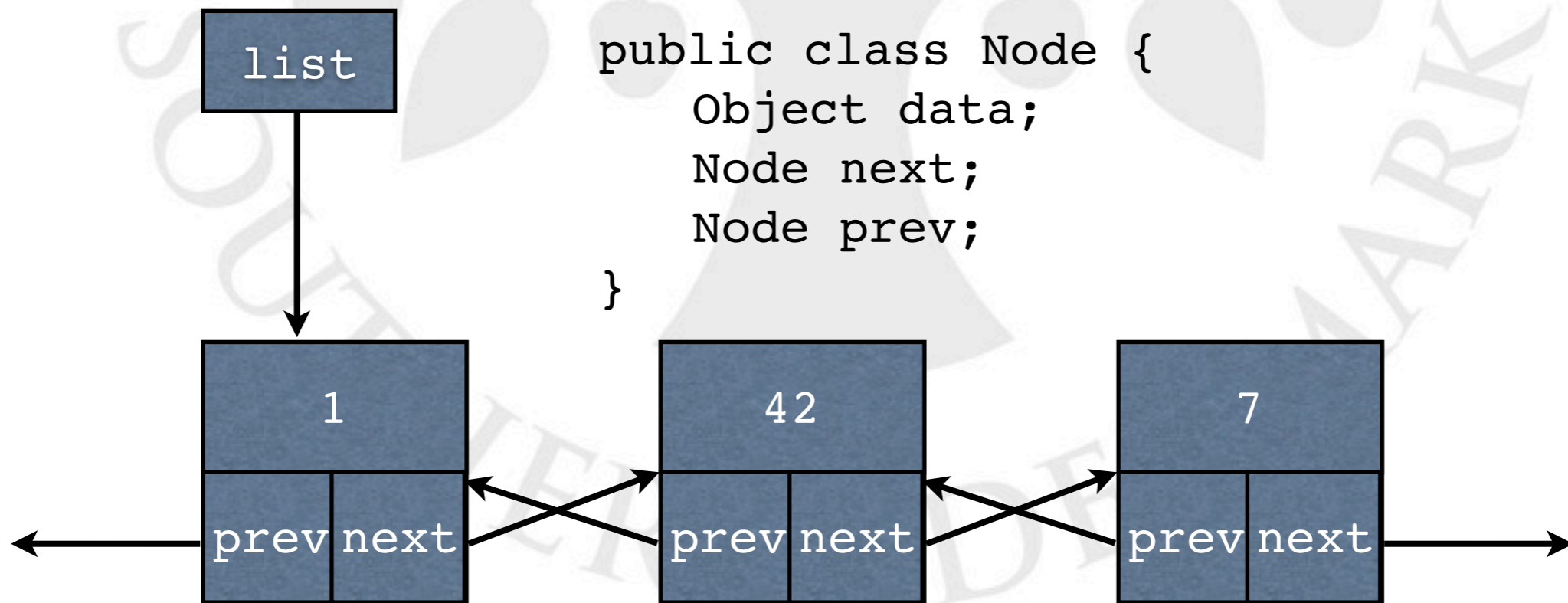
# Lister

- Dobbelt-hægtet liste
  - Ligesom enkelt-hægtet liste, men med et ekstra felt
  - Lige så effektiv som enkelt-hægtet liste
  - Kan nu løbe igennem begge veje
- `java.util.LinkedList`
  - Dobbelt-hægtet liste



# Lister

- Dobbelt-hægtet liste
  - Ligesom enkelt-hægtet liste, men med et ekstra felt
  - Lige så effektiv som enkelt-hægtet liste
  - Kan nu løbe igennem begge veje
- `java.util.LinkedList`
  - Dobbelt-hægtet liste
  - Implementerer interfacet `java.util.List`



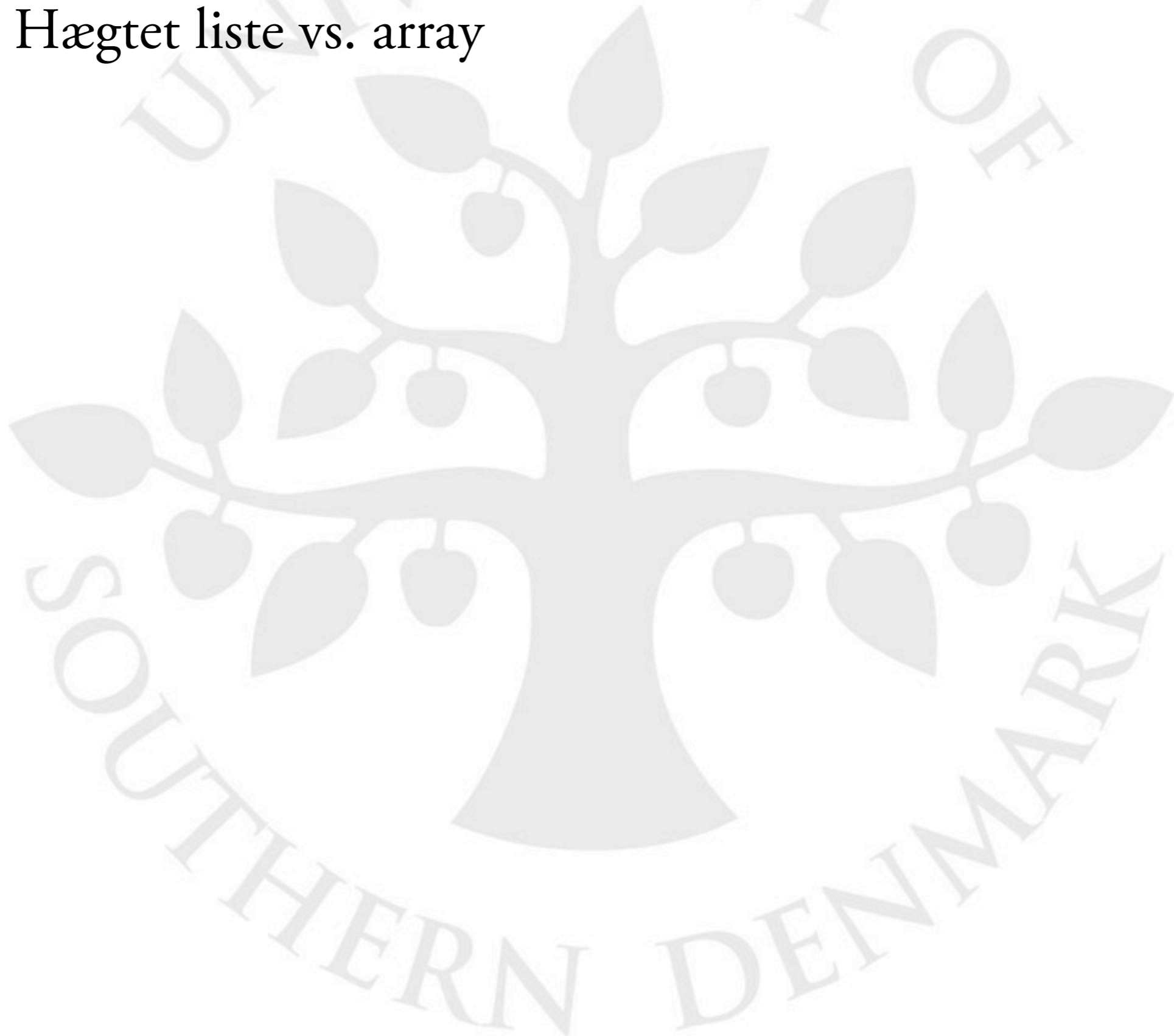


# Lister



# Lister

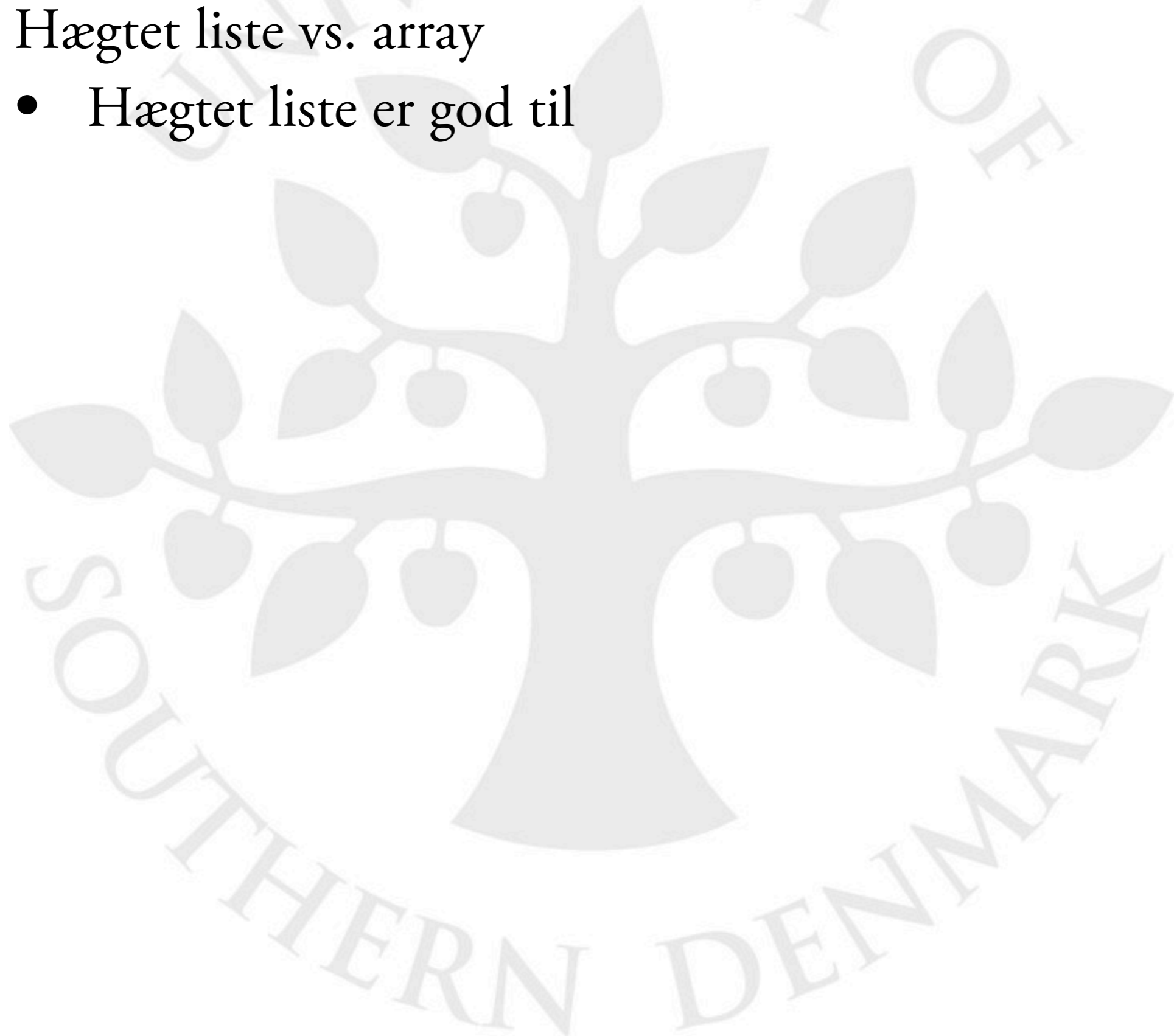
- Hægtet liste vs. array





# Lister

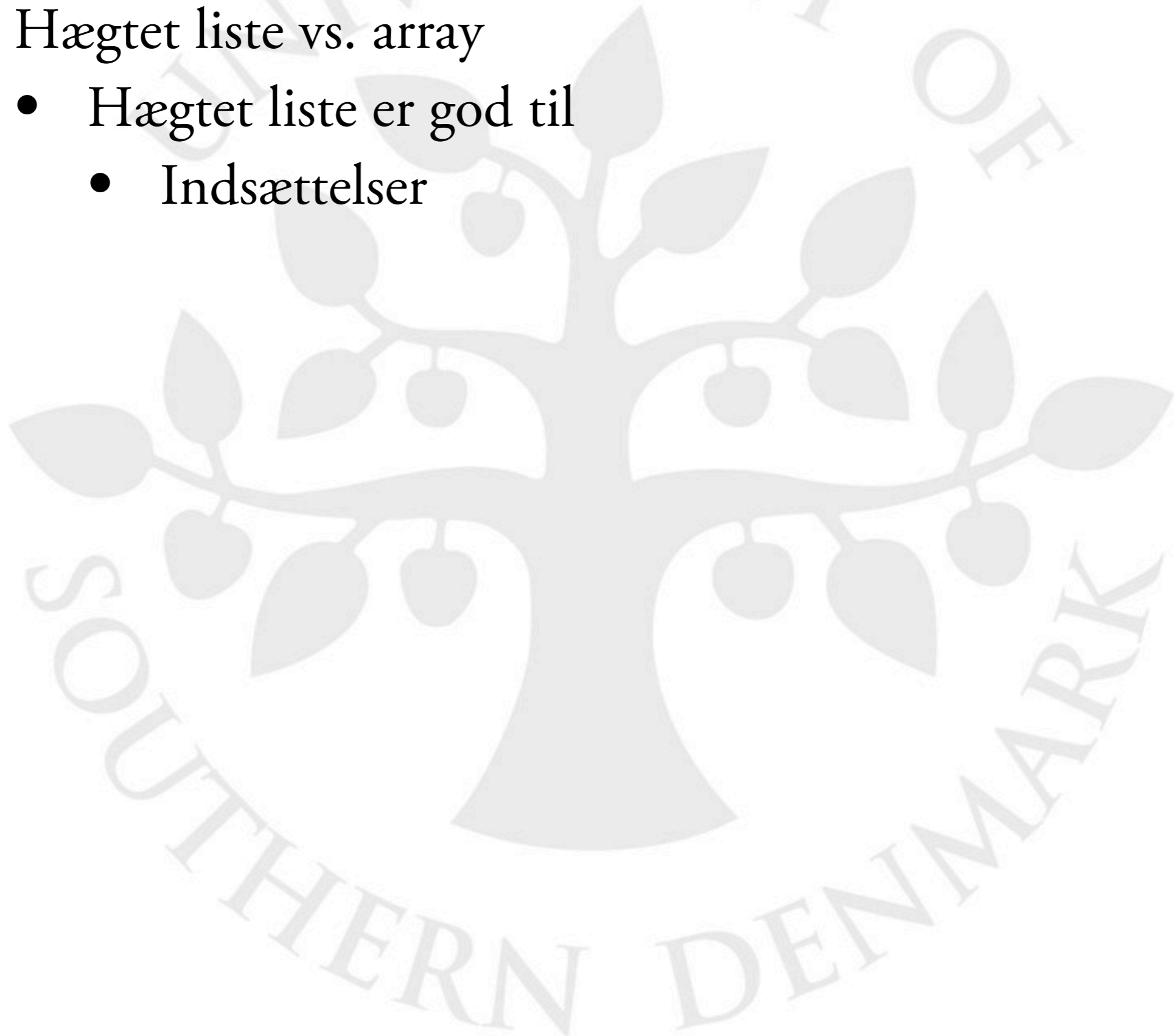
- Hægtet liste vs. array
  - Hægtet liste er god til





# Lister

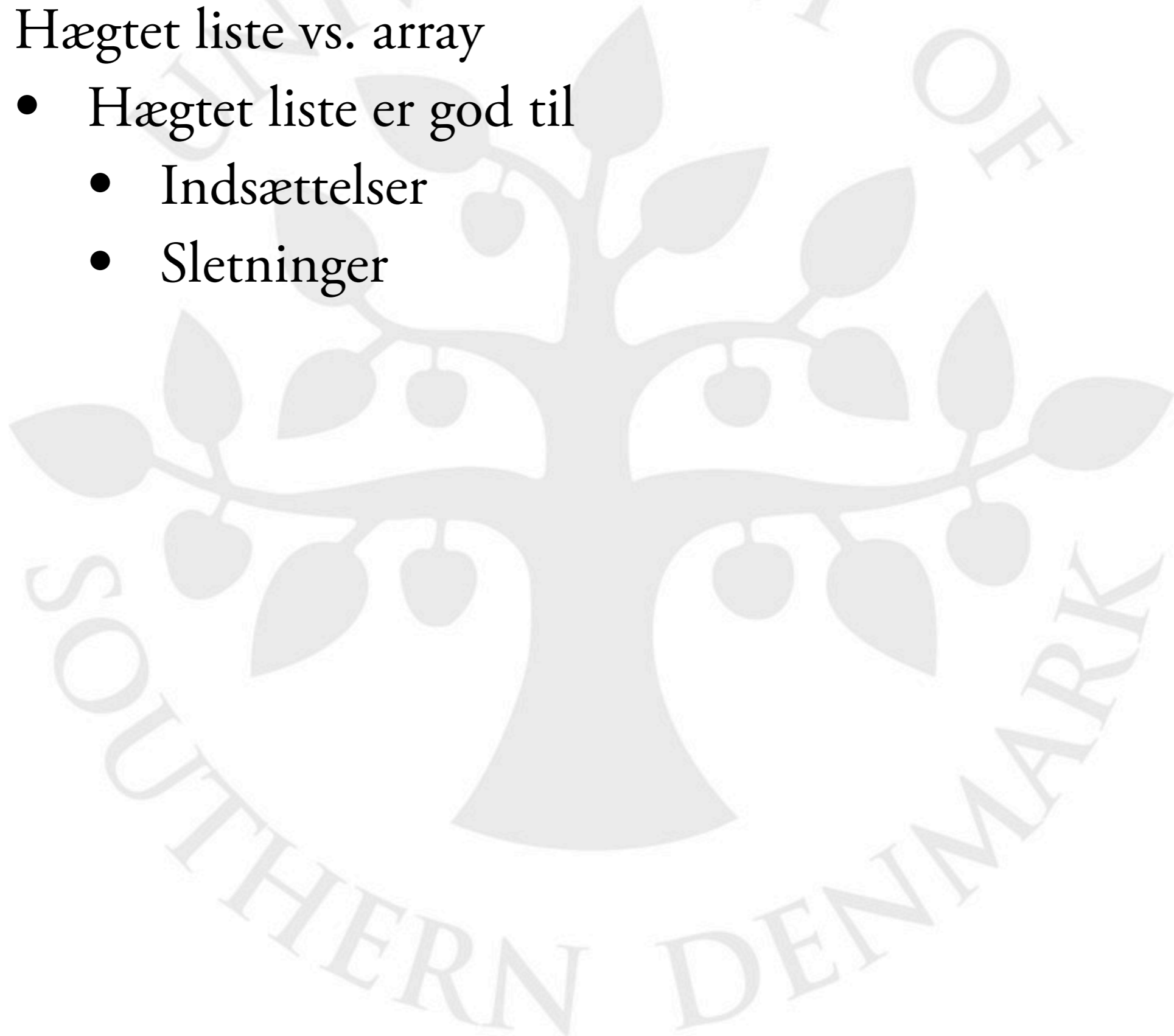
- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser





# Lister

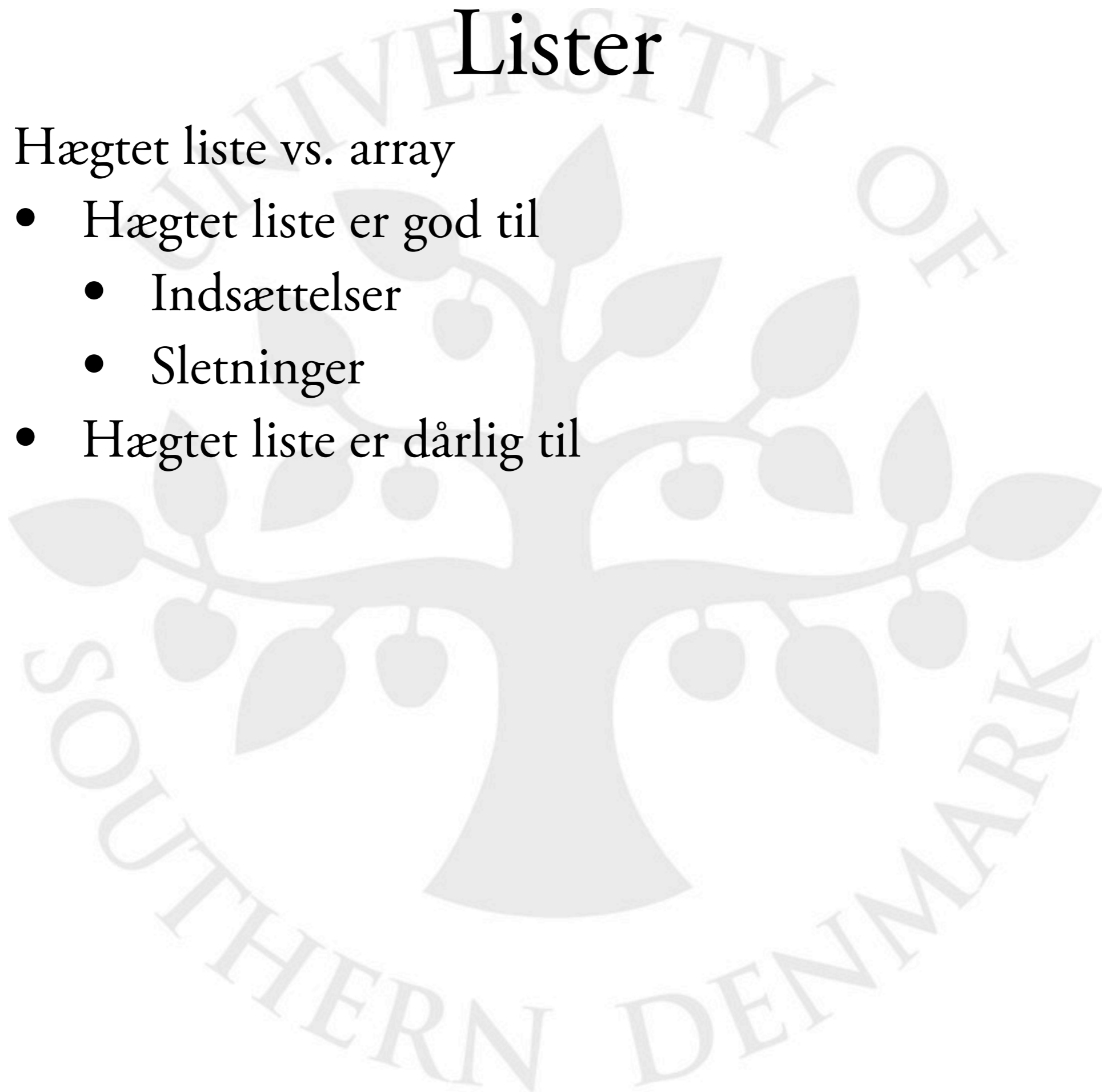
- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger





# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til





# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til
    - Indeksering

# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til
    - Indeksering
  - Array er god til



# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til
    - Indeksering
  - Array er god til
    - Indeksering



# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til
    - Indeksering
- Array er god til
  - Indeksering
- Array er dårlig til



# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til
    - Indeksering
  - Array er god til
    - Indeksering
  - Array er dårlig til
    - Indsættelser

# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til
    - Indeksering
  - Array er god til
    - Indeksering
  - Array er dårlig til
    - Indsættelser
    - Sletninger

# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til
    - Indeksering
  - Array er god til
    - Indeksering
  - Array er dårlig til
    - Indsættelser
    - Sletninger
- Hmm... kan man mon være god til det hele?

# Lister

- Hægtet liste vs. array
  - Hægtet liste er god til
    - Indsættelser
    - Sletninger
  - Hægtet liste er dårlig til
    - Indeksering
  - Array er god til
    - Indeksering
  - Array er dårlig til
    - Indsættelser
    - Sletninger
- Hmm... kan man mon være god til det hele?
  - Ikke helt, men næsten

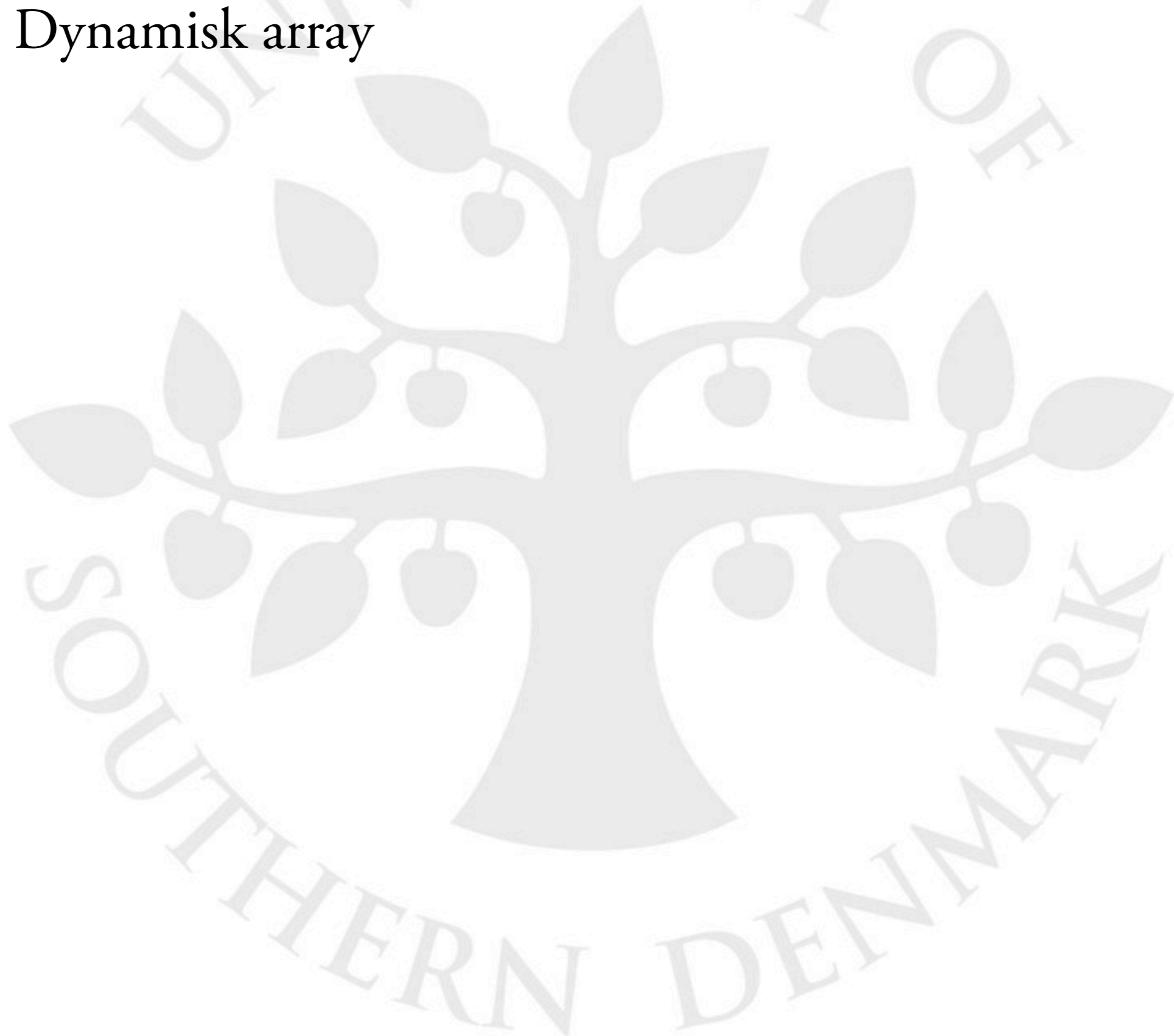
# Lister





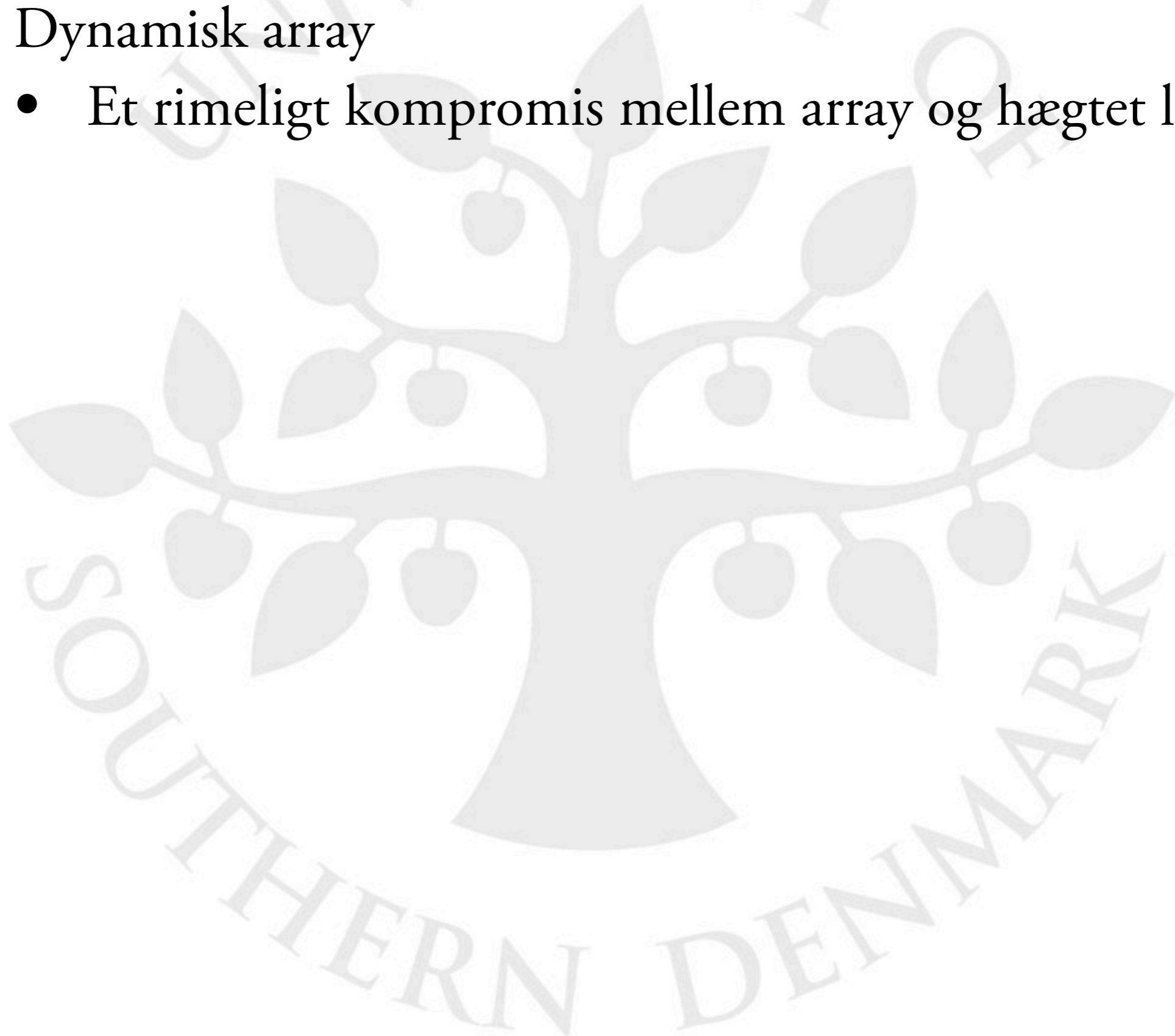
# Lister

- Dynamisk array



# Lister

- Dynamisk array
  - Et rimeligt kompromis mellem array og hæftet liste



# Lister

- Dynamisk array
  - Et rimeligt kompromis mellem array og hæftet liste
  - “Array” hvor der kan indsættes vilkårlig ofte





# Lister

- Dynamisk array
  - Et rimeligt kompromis mellem array og hæftet liste
  - “Array” hvor der kan indsættes vilkårlig ofte
  - Udgangspunktet er et almindeligt array



# Lister

- Dynamisk array
  - Et rimeligt kompromis mellem array og hæftet liste
  - “Array” hvor der kan indsættes vilkårlig ofte
  - Udgangspunktet er et almindeligt array
  - Implementeret i `java.util.ArrayList`

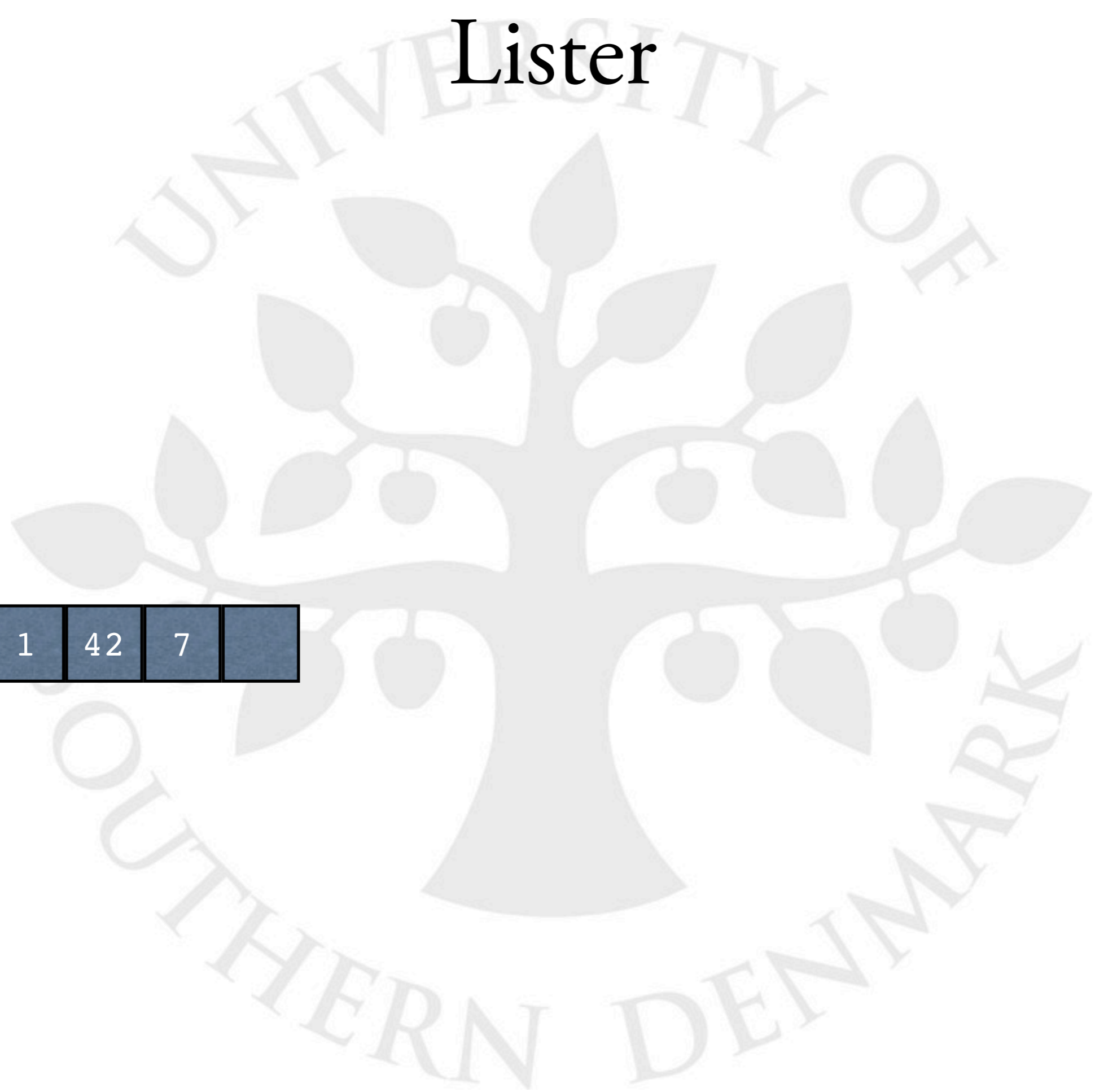


# Lister

- Dynamisk array
  - Et rimeligt kompromis mellem array og hæftet liste
  - “Array” hvor der kan indsættes vilkårlig ofte
  - Udgangspunktet er et almindeligt array
  - Implementeret i `java.util.ArrayList`
    - Implementerer også interfacet `java.util.List`

# Lister

|   |    |   |  |
|---|----|---|--|
| 1 | 42 | 7 |  |
|---|----|---|--|



# Lister

- Indsættelse i dynamisk array

|   |    |   |  |
|---|----|---|--|
| 1 | 42 | 7 |  |
|---|----|---|--|



# Lister

- Indsættelse i dynamisk array
- Sekvens af operationer

|   |    |   |  |
|---|----|---|--|
| 1 | 42 | 7 |  |
|---|----|---|--|



# Lister

- Indsættelse i dynamisk array
- Sekvens af operationer
  - `add(9)`

|   |    |   |  |
|---|----|---|--|
| 1 | 42 | 7 |  |
|---|----|---|--|



# Lister

- Indsættelse i dynamisk array
- Sekvens af operationer
  - `add(9)`

|   |    |   |   |
|---|----|---|---|
| 1 | 42 | 7 | 9 |
|---|----|---|---|





# Lister

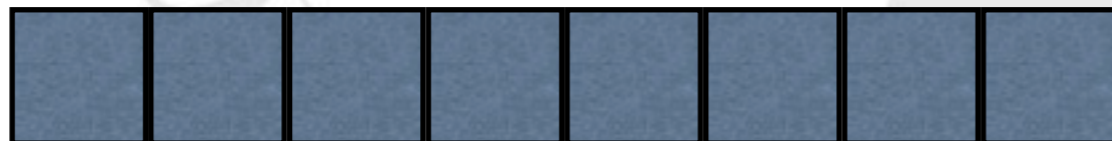
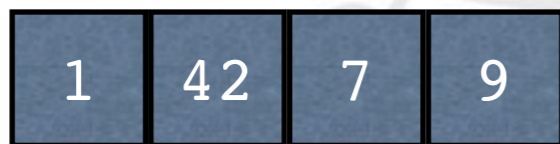
- Indsættelse i dynamisk array
- Sekvens af operationer
  - `add(9)`
  - `add(11)`

|   |    |   |   |
|---|----|---|---|
| 1 | 42 | 7 | 9 |
|---|----|---|---|



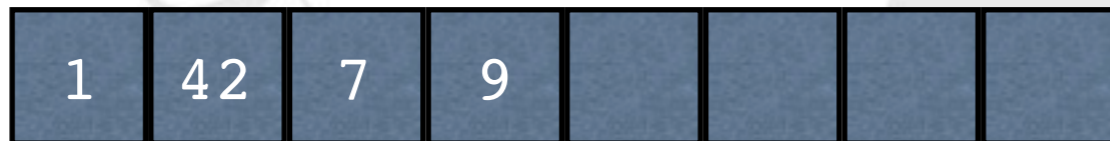
# Lister

- Indsættelse i dynamisk array
- Sekvens af operationer
  - `add(9)`
  - `add(11)`



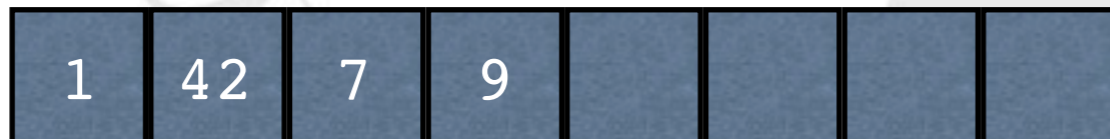
# Lister

- Indsættelse i dynamisk array
- Sekvens af operationer
  - `add(9)`
  - `add(11)`



# Lister

- Indsættelse i dynamisk array
- Sekvens af operationer
  - `add(9)`
  - `add(11)`



# Lister

- Indsættelse i dynamisk array
- Sekvens af operationer
  - `add(9)`
  - `add(11)`



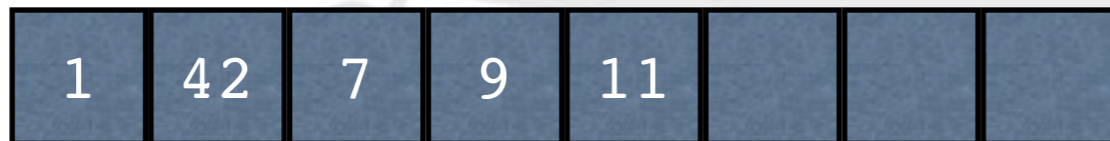
# Lister

|   |    |   |   |    |  |  |  |
|---|----|---|---|----|--|--|--|
| 1 | 42 | 7 | 9 | 11 |  |  |  |
|---|----|---|---|----|--|--|--|



# Lister

- Sletning i dynamisk array



|   |    |   |   |    |  |  |  |
|---|----|---|---|----|--|--|--|
| 1 | 42 | 7 | 9 | 11 |  |  |  |
|---|----|---|---|----|--|--|--|

# Lister

- Sletning i dynamisk array
- Sekvens af operationer

|   |    |   |   |    |  |  |  |
|---|----|---|---|----|--|--|--|
| 1 | 42 | 7 | 9 | 11 |  |  |  |
|---|----|---|---|----|--|--|--|





# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`

|   |    |   |   |    |  |  |  |
|---|----|---|---|----|--|--|--|
| 1 | 42 | 7 | 9 | 11 |  |  |  |
|---|----|---|---|----|--|--|--|

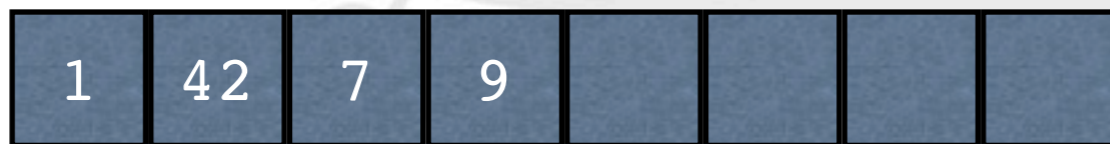
# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`

|   |    |   |   |  |  |  |  |
|---|----|---|---|--|--|--|--|
| 1 | 42 | 7 | 9 |  |  |  |  |
|---|----|---|---|--|--|--|--|

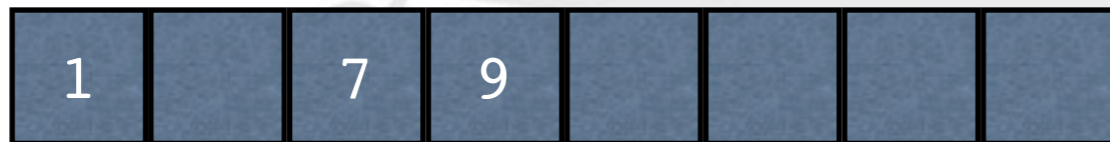
# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`



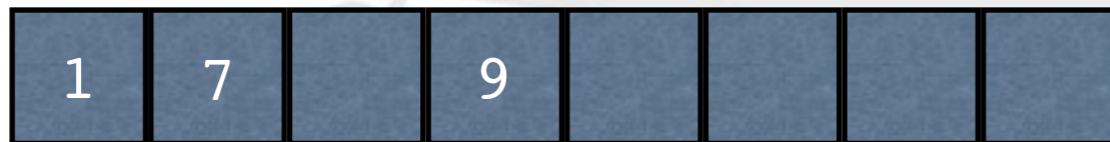
# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`



# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`



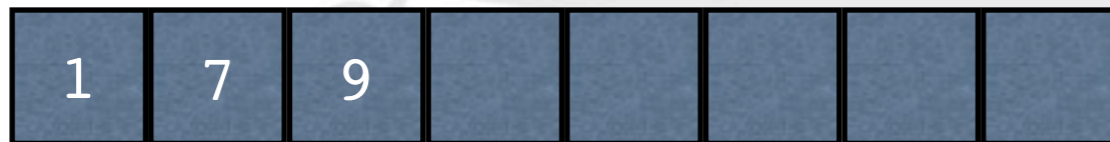
# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`



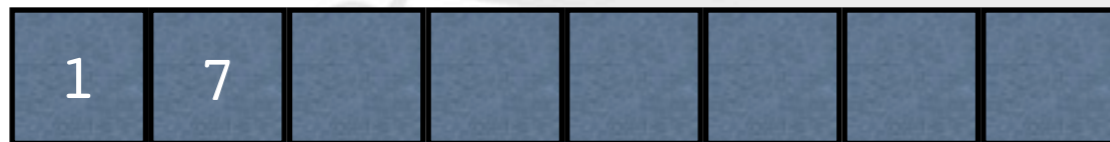
# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`
  - `remove(9)`



# Lister

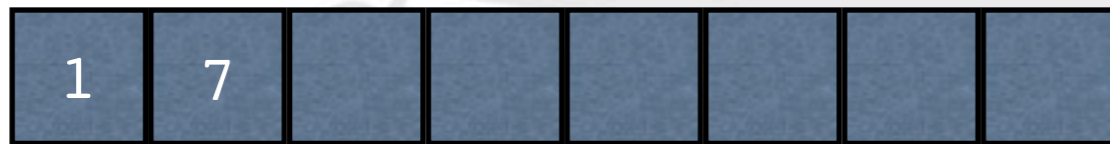
- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`
  - `remove(9)`





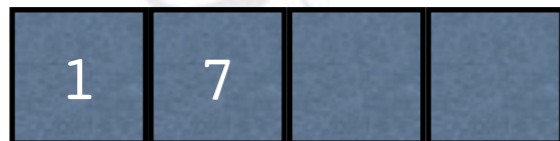
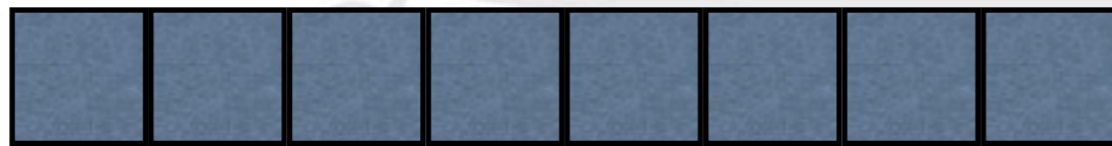
# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`
  - `remove(9)`



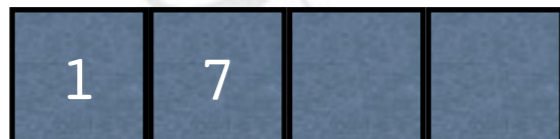
# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`
  - `remove(9)`



# Lister

- Sletning i dynamisk array
- Sekvens af operationer
  - `remove(11)`
  - `remove(42)`
  - `remove(9)`





# Lister





# Lister

- Andre operationer på dynamisk array



# Lister

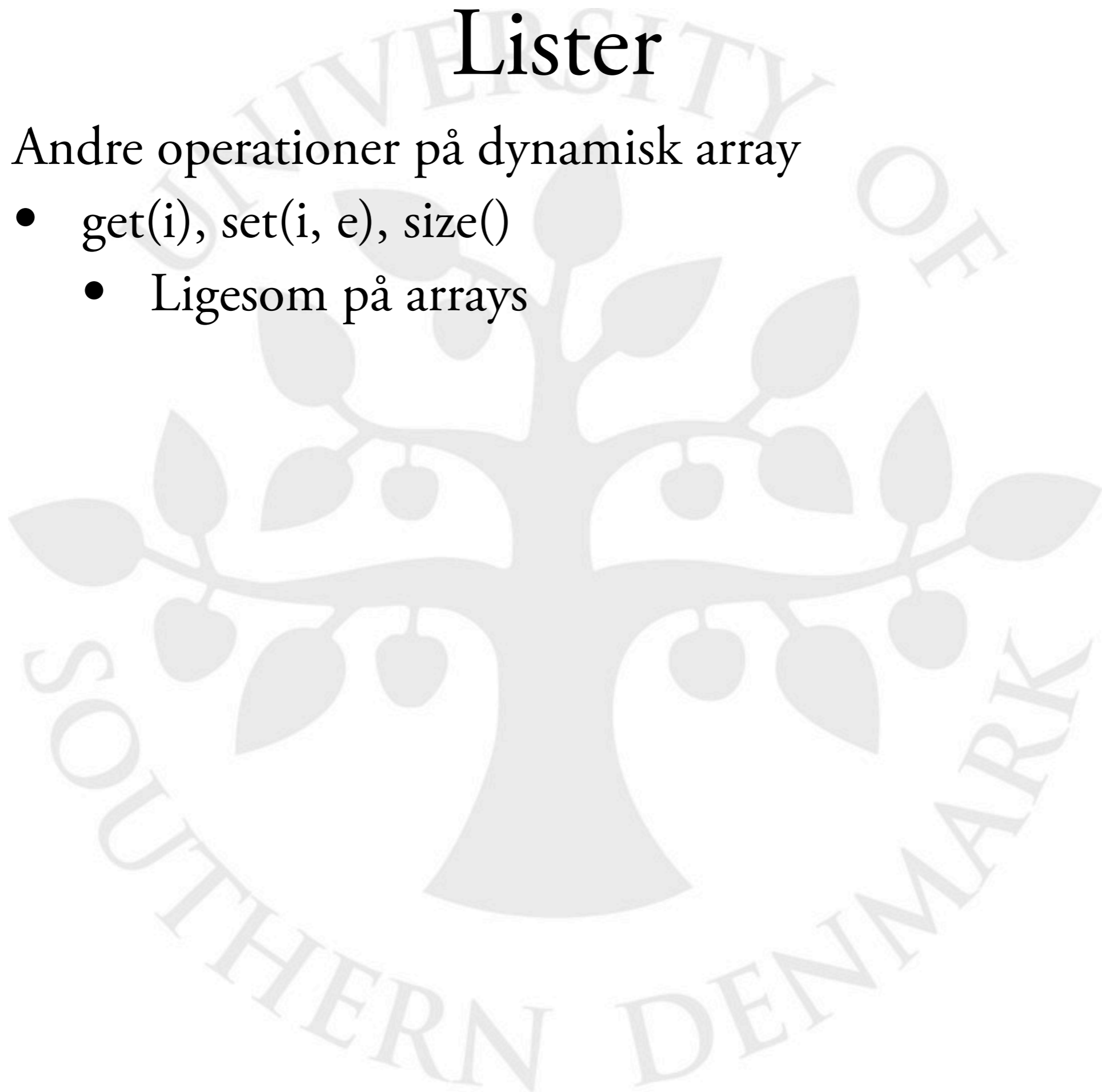
- Andre operationer på dynamisk array
  - `get(i)`, `set(i, e)`, `size()`





# Lister

- Andre operationer på dynamisk array
  - `get(i)`, `set(i, e)`, `size()`
    - Ligesom på arrays



# Lister

- Andre operationer på dynamisk array
  - `get(i)`, `set(i, e)`, `size()`
    - Ligesom på arrays
  - Kan også have `add(i, e)` som indsætter et element midt i listen





# Lister

- Andre operationer på dynamisk array
  - `get(i)`, `set(i, e)`, `size()`
    - Ligesom på arrays
  - Kan også have `add(i, e)` som indsætter et element midt i listen
    - Flytter elementer så der er plads (ligesom `remove(e)` gør)

# Lister

- Andre operationer på dynamisk array
  - `get(i)`, `set(i, e)`, `size()`
    - Ligesom på arrays
  - Kan også have `add(i, e)` som indsætter et element midt i listen
    - Flytter elementer så der er plads (ligesom `remove(e)` gør)
- Bemærk

# Lister

- Andre operationer på dynamisk array
  - `get(i)`, `set(i, e)`, `size()`
    - Ligesom på arrays
  - Kan også have `add(i, e)` som indsætter et element midt i listen
    - Flytter elementer så der er plads (ligesom `remove(e)` gør)
- Bemærk
  - `add(e)` og `remove(e)` kan tage lang tid nogle gange

# Lister

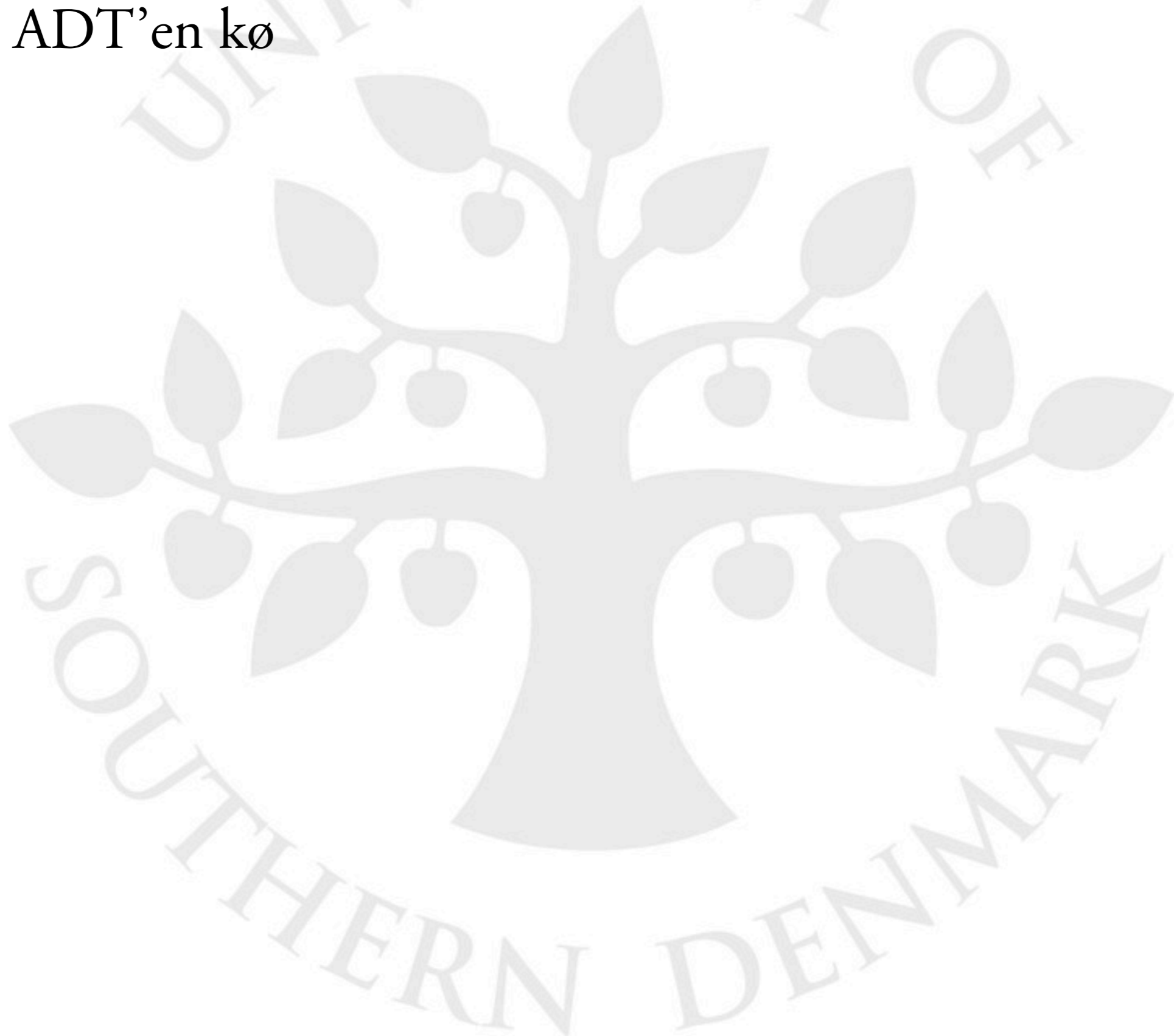
- Andre operationer på dynamisk array
  - `get(i)`, `set(i, e)`, `size()`
    - Ligesom på arrays
  - Kan også have `add(i, e)` som indsætter et element midt i listen
    - Flytter elementer så der er plads (ligesom `remove(e)` gør)
- Bemærk
  - `add(e)` og `remove(e)` kan tage lang tid nogle gange
    - Bare ikke “særligt tit”

# Køer



# Køer

- ADT'en kø



# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge



# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out





# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende



# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden



# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden
- Data



# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden
- Data
  - Alt - ingen restriktioner



# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden
- Data
  - Alt - ingen restriktioner
- Operationer



# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden
- Data
  - Alt - ingen restriktioner
- Operationer
  - enqueue(e)

# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden
- Data
  - Alt - ingen restriktioner
- Operationer
  - enqueue(e)
  - dequeue()



# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden
- Data
  - Alt - ingen restriktioner
- Operationer
  - enqueue(e)
  - dequeue()
  - front()



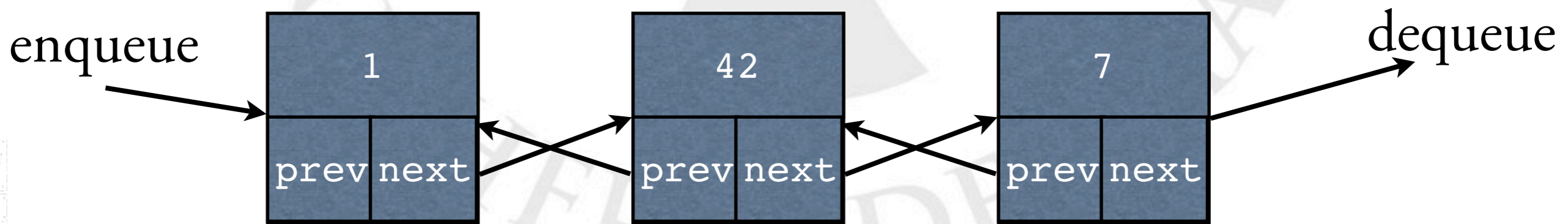
# Køer

- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden
- Data
  - Alt - ingen restriktioner
- Operationer
  - enqueue(e)
  - dequeue()
  - front()
  - empty()

# Køer

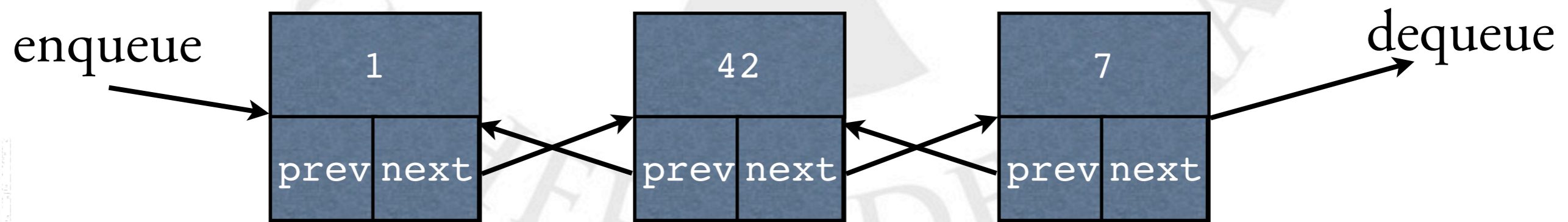
- ADT'en kø
  - En indskrænkning af ADT'en følge
  - FIFO - First In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den anden
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - enqueue(e)
    - dequeue()
    - front()
    - empty()
  - Se `java.util.Deque` interfacet

# Køer



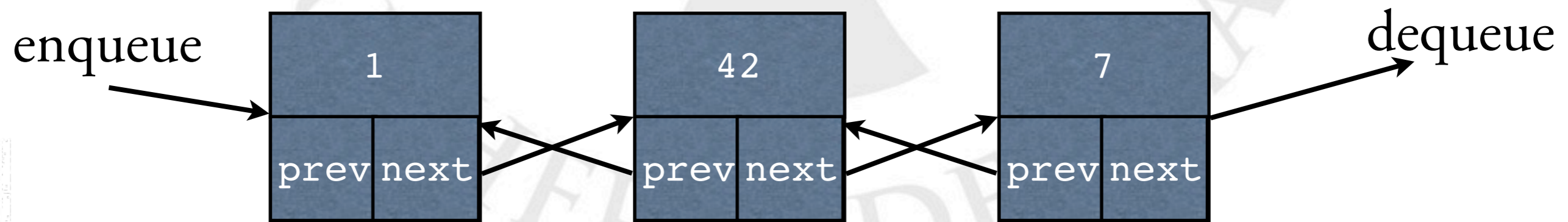
# Køer

- Design og implementation?



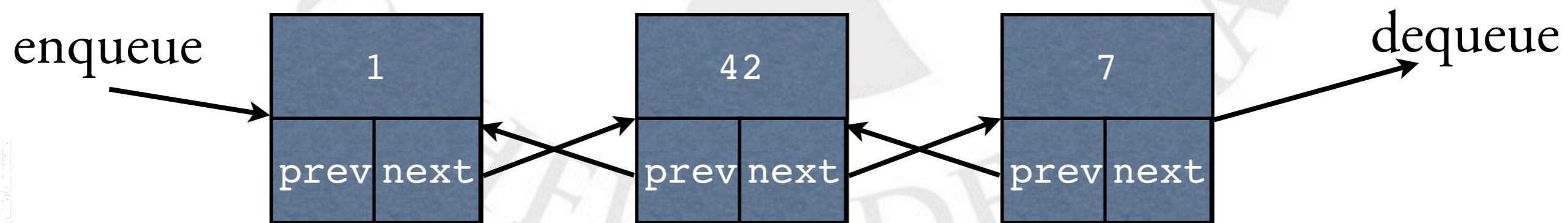
# Køer

- Design og implementation?
  - Hægtet liste



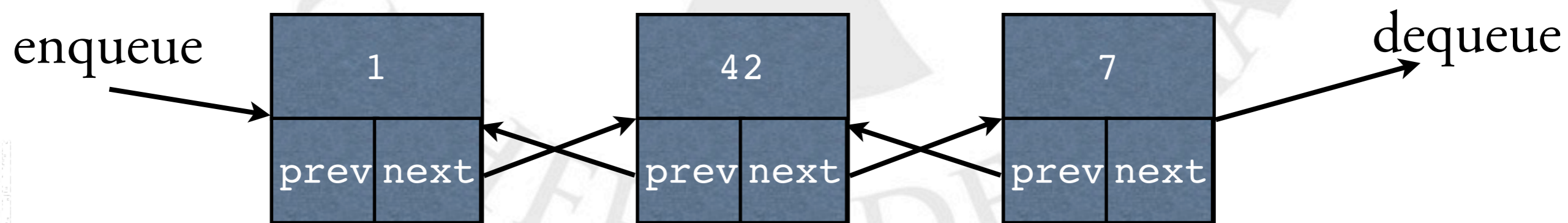
# Køer

- Design og implementation?
  - Hægtet liste
    - Enqueue(e)



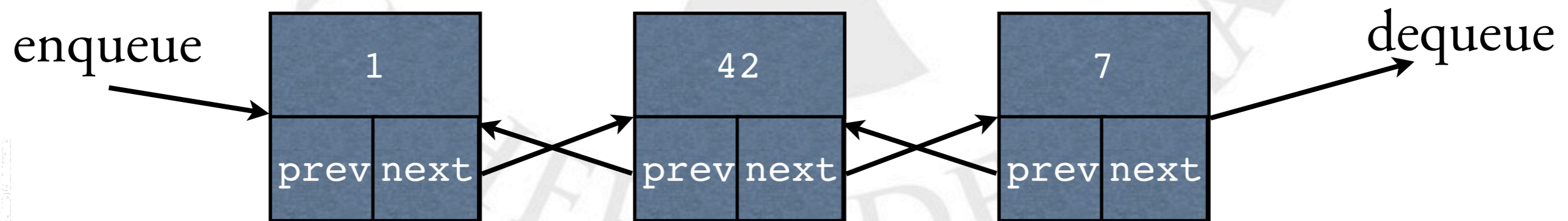
# Køer

- Design og implementation?
  - Hægtet liste
    - Enqueue(e)
      - Indsæt elementet forrest i listen



# Køer

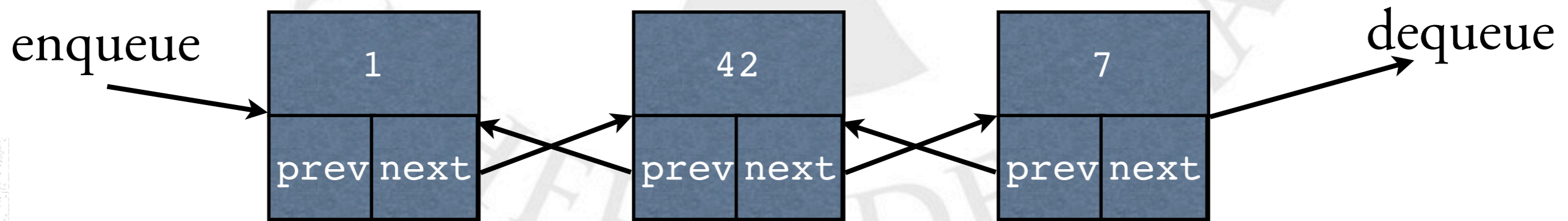
- Design og implementation?
  - Hægtet liste
    - Enqueue(e)
      - Indsæt elementet forrest i listen
    - Dequeue()





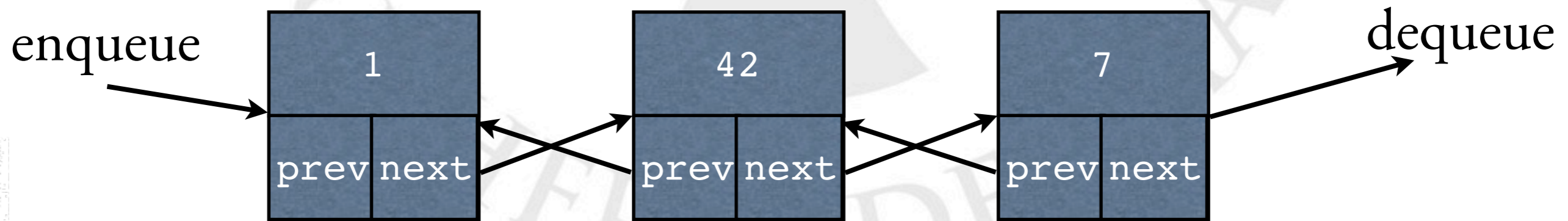
# Køer

- Design og implementation?
  - Hægtet liste
    - Enqueue(e)
      - Indsæt elementet forrest i listen
    - Dequeue()
      - Udtag det bagereste element i listen



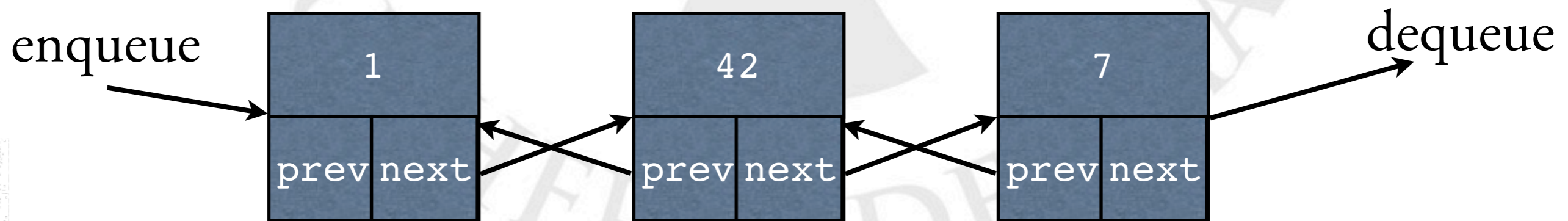
# Køer

- Design og implementation?
  - Hægtet liste
    - Enqueue(e)
      - Indsæt elementet forrest i listen
    - Dequeue()
      - Udtag det bagereste element i listen
  - Alle operationer er effektive



# Køer

- Design og implementation?
  - Hægtet liste
    - Enqueue(e)
      - Indsæt elementet forrest i listen
    - Dequeue()
      - Udtag det bagereste element i listen
  - Alle operationer er effektive
  - `java.util.LinkedList` implementerer interfacet `java.util.Deque`

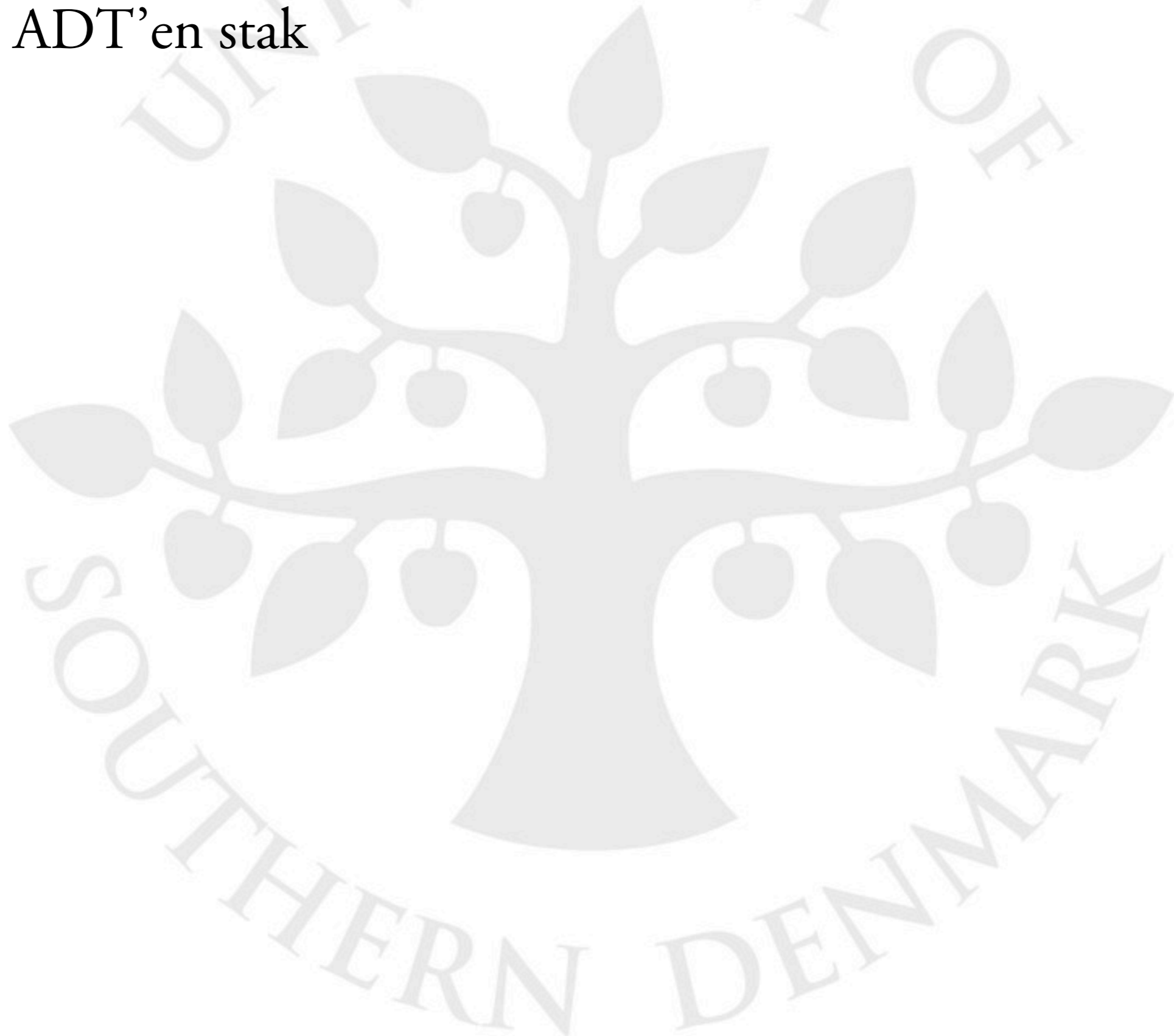


# Stakke



# Stakke

- ADT'en stak



# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge



# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out



# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende





# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende



# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende
- Data



# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende
- Data
  - Alt - ingen restriktioner



# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende
  - Data
    - Alt - ingen restriktioner
  - Operationer

# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - `push(e)`



# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende
- Data
  - Alt - ingen restriktioner
- Operationer
  - `push(e)`
  - `pop()`



# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende
- Data
  - Alt - ingen restriktioner
- Operationer
  - push(e)
  - pop()
  - peek()

# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende
- Data
  - Alt - ingen restriktioner
- Operationer
  - push(e)
  - pop()
  - peek()
  - empty()





# Stakke

- ADT'en stak
  - En indskrænkning af ADT'en følge
  - LIFO - Last In First Out
    - Indsætte elementer i den ene ende
    - Tage elementer ud i den samme ende
  - Data
    - Alt - ingen restriktioner
  - Operationer
    - `push(e)`
    - `pop()`
    - `peek()`
    - `empty()`
  - Se `java.util.Deque` interfacet



# Stakke



# Stakke

- Design og implementation?



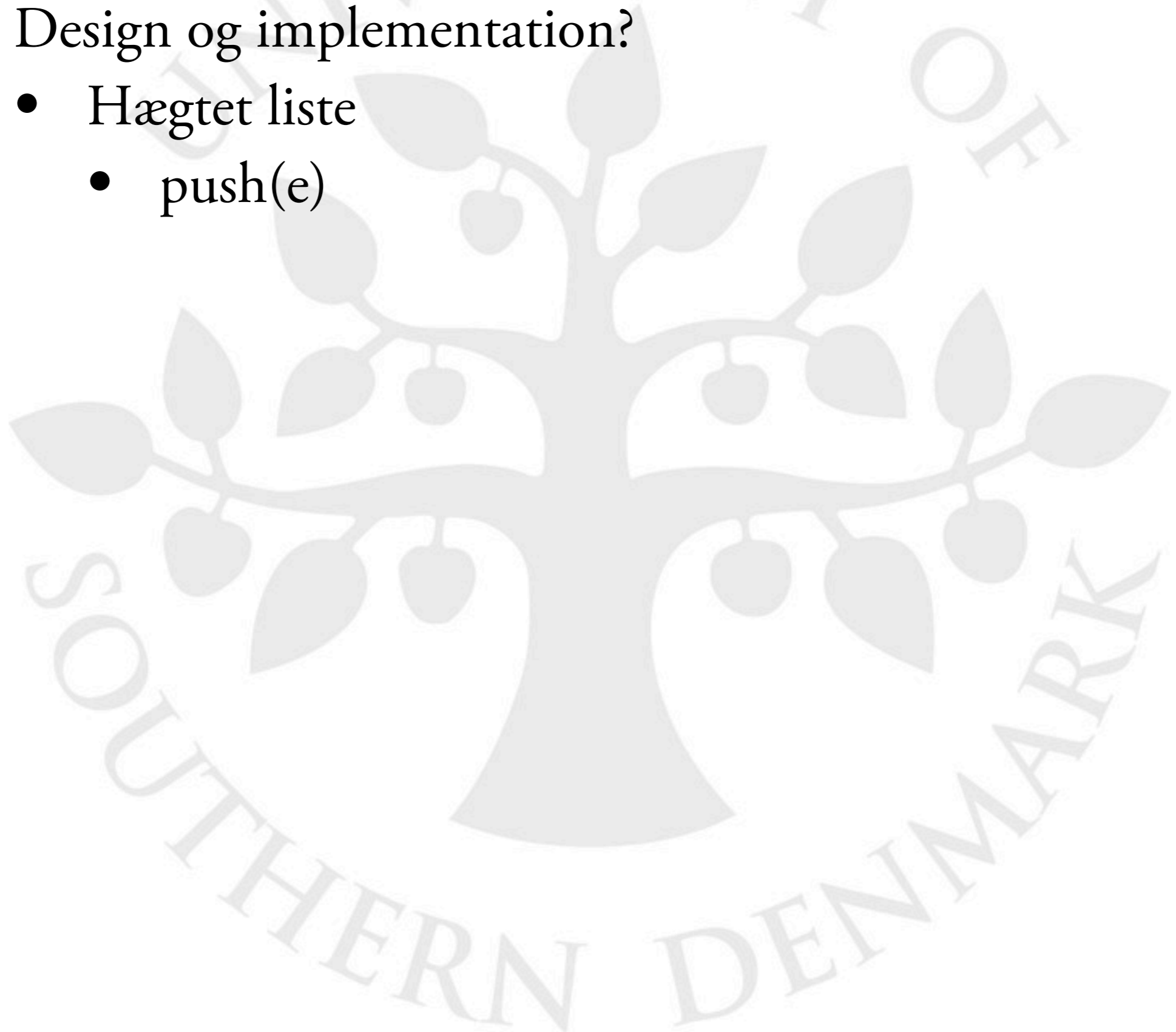
# Stakke

- Design og implementation?
  - Hægtet liste



# Stakke

- Design og implementation?
  - Hægtet liste
    - `push(e)`



# Stakke

- Design og implementation?
  - Hægtet liste
    - push(e)
      - Indsæt elementet forrest i listen



# Stakke

- Design og implementation?
  - Hægtet liste
    - push(e)
      - Indsæt elementet forrest i listen
    - pop()



# Stakke

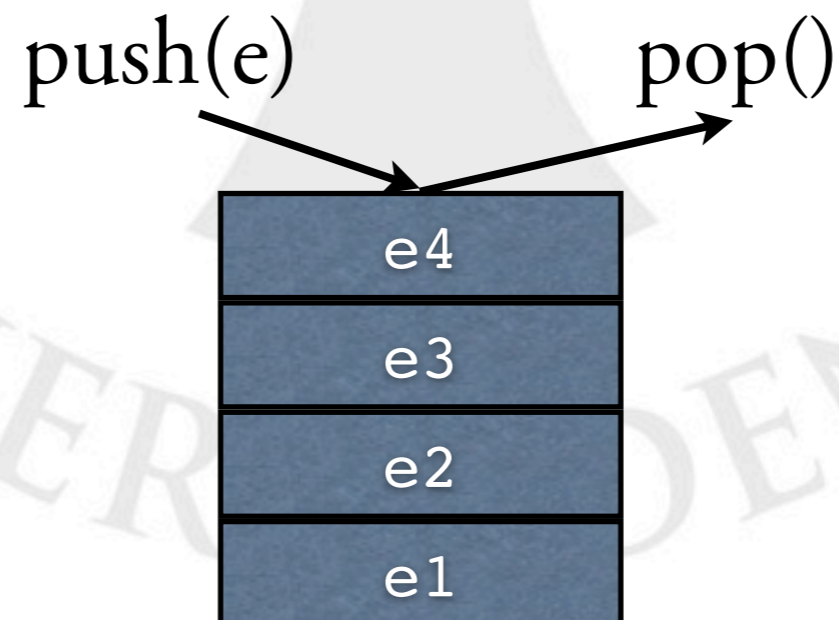
- Design og implementation?
  - Hægtet liste
    - push(e)
      - Indsæt elementet forrest i listen
    - pop()
      - Udtag det forreste element i listen





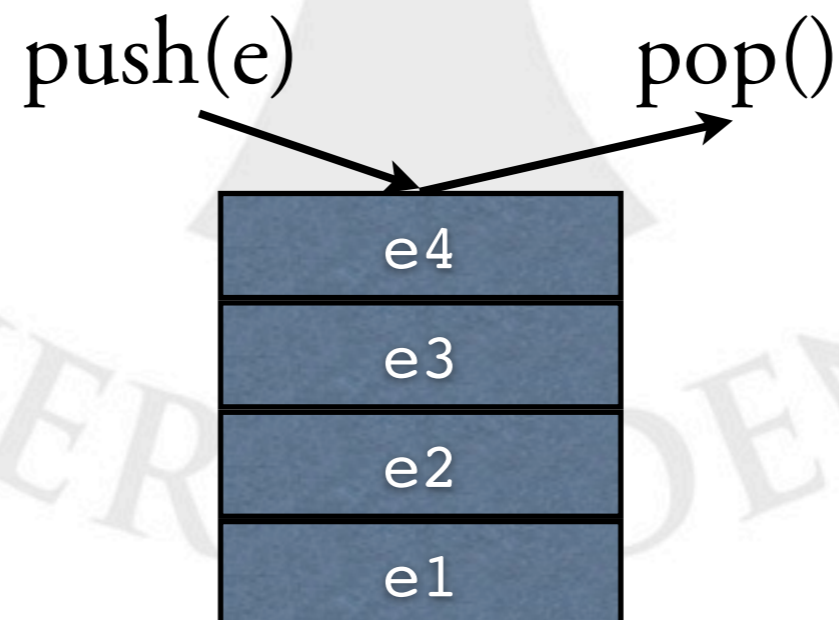
# Stakke

- Design og implementation?
  - Hægtet liste
    - `push(e)`
      - Indsæt elementet forrest i listen
    - `pop()`
      - Udtag det forreste element i listen



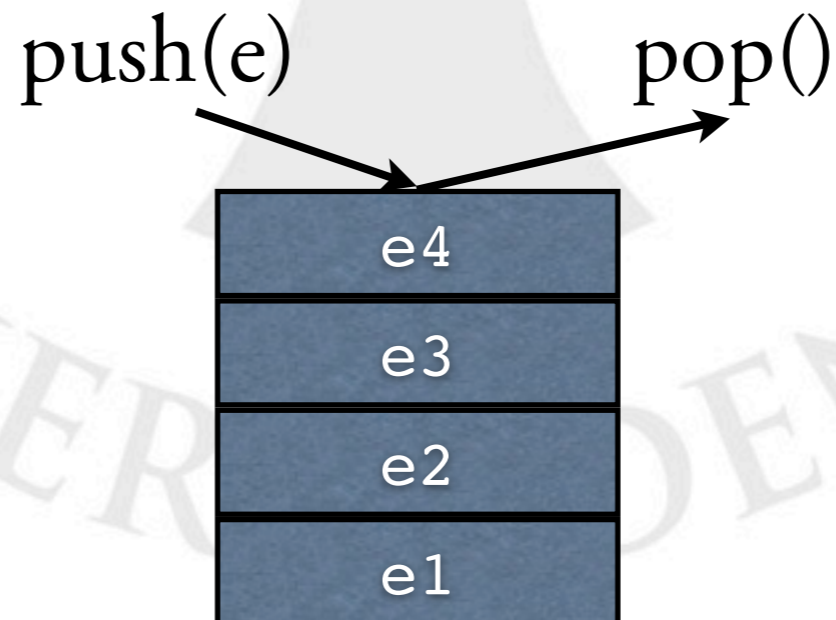
# Stakke

- Design og implementation?
  - Hægtet liste
    - `push(e)`
      - Indsæt elementet forrest i listen
    - `pop()`
      - Udtag det forreste element i listen
  - Alle operationer er effektive



# Stakke

- Design og implementation?
  - Hægtet liste
    - `push(e)`
      - Indsæt elementet forrest i listen
    - `pop()`
      - Udtag det forreste element i listen
  - Alle operationer er effektive
  - `java.util.LinkedList` implementerer interfacet `java.util.Deque`



# Stakke & Køer

- Bemærk
  - `java.util.Stack`
    - Lad være med at bruge den
  - `java.util.ArrayDeque`
    - Implementerer `Deque` interfacet
    - Både stak og kø
    - Opfører sig internt som `ArrayList`
    - Java (Sun) siger
      - “This class is likely to be faster than `Stack` when used as a stack, and faster than `LinkedList` when used as a queue.”