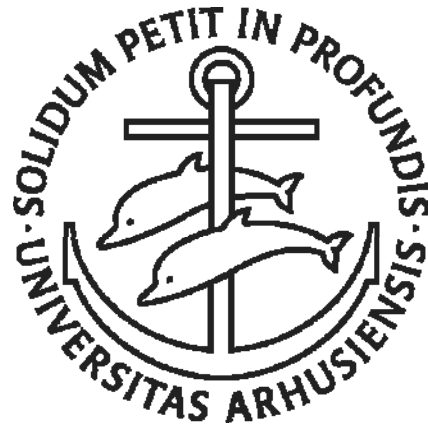


# Algoritmer og Datastrukturer 1

Union-Find [CLRS, kapitel 21.2-21.3]



**Gerth Stølting Brodal**

Aarhus Universitet

# Union-Find

**MakeSet( $S, x$ )**

Opret en ny mængde  $\{ x \}$

**Union( $x, y$ )**

Erstat  $S_x = \{ \dots, x, \dots \}$   $S_y = \{ \dots, x, \dots \}$

med  $S_x \cup S_y = \{ \dots, x, \dots, y, \dots \}$

**FindSet( $x$ )**

Retuner en *repræsentant* for  $S_x = \{ \dots, x, \dots \}$

FindSet( $x$ )=FindSet( $y$ ) hvis og kun hvis  $x$   
og  $y$  er i samme mængde

# Trærepræsentation (I)

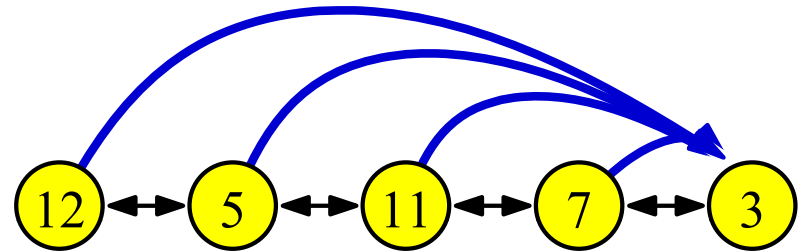
- Mængde = enkelt kædet liste
- MakeSet = lav en ny knude
- FindSet = returner sidste knude
- Union = konkatener listerne



<b>MakeSet(<math>S,x</math>)</b>	$O(1)$
<b>Union(<math>x,y</math>)</b>	$O( S_x + S_y )$
<b>FindSet(<math>x</math>)</b>	$O( S_x )$

# Trærepræsentation (II)

- Mængde = enkelt kædet liste
- MakeSet = lav en ny knude
- FindSet = returner sidste knude
- Union = konkatener listerne og opdater pointerne



<b>MakeSet(<math>S,x</math>)</b>	$O(1)$
<b>Union(<math>x,y</math>)</b>	$O(\min( S_x , S_y ))$
<b>FindSet(<math>x</math>)</b>	$O(1)$

# Sekvens af Union

## Sætning

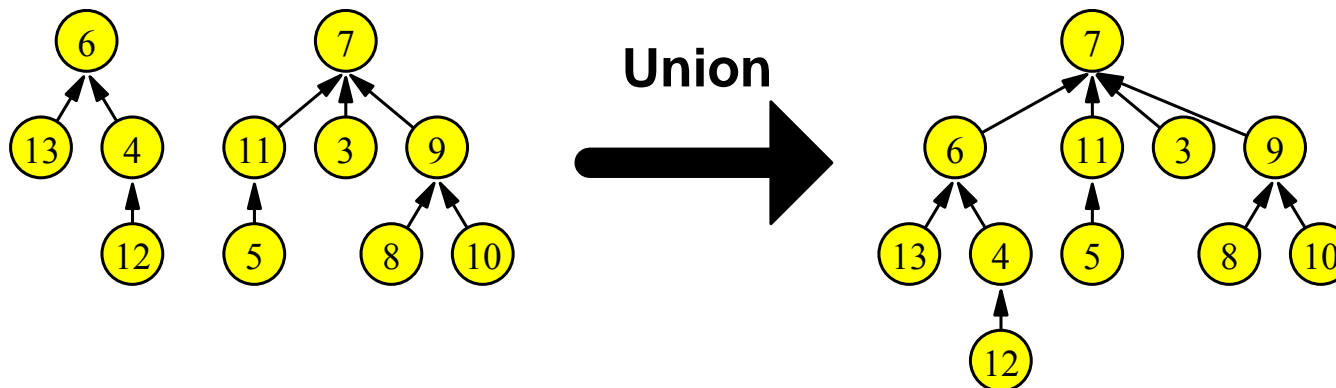
Et sekvens af  $n$  union operationer tager tid højst  $O(n \cdot \log n)$

## Bevis

Hver pointer flyttes højst  $\log n$  gange  
(hver gang til en liste der mindst er dobbelt så stor)

# Trærepræsentation

- Mængde = træ
- MakeSet = lav en ny knude
- FindSet = returner roden
- Union = Sæt det "lille" træ under roden af det "store" træ



# Stikomprimering

MAKE-SET( $x$ )

- 1  $p[x] \leftarrow x$
- 2  $rank[x] \leftarrow 0$

UNION( $x, y$ )

- 1 LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))

LINK( $x, y$ )

- 1 **if**  $rank[x] > rank[y]$
- 2     **then**  $p[y] \leftarrow x$
- 3     **else**  $p[x] \leftarrow y$
- 4         **if**  $rank[x] = rank[y]$
- 5             **then**  $rank[y] \leftarrow rank[y] + 1$

FIND-SET( $x$ )

- 1 **if**  $x \neq p[x]$
- 2     **then**  $p[x] \leftarrow$  FIND-SET( $p[x]$ )
- 3 **return**  $p[x]$

stikomprimering

# Analyse af Trærepræsentation med Rank-Linkning

## Lemma

$$height[x] \leq rank[x]$$

$$size[x] \geq 2^{rank[x]}$$

## Bevis

Induktion.

<b>MakeSet(<math>S, x</math>)</b>	$O(1)$
<b>Union(<math>x, y</math>)</b>	$O((\log  S_x ) + (\log  S_y ))$
<b>FindSet(<math>x</math>)</b>	$O(\log  S_x )$

(Ovenstående udnytter kun linking by rank)



# Analyse af Trærepræsentation med Stikomprimering

## Sætning

En sekvens af  $m$  Union-Find operation på  $n$  elementer tager tid  $O(m \cdot \alpha(n))$  hvor  $\alpha(n)$  er den Inverse til Ackerman funktionen ([CLRS], kapitel 21.4) som for alle praktiske formål  $\alpha(n) \leq 4$ .