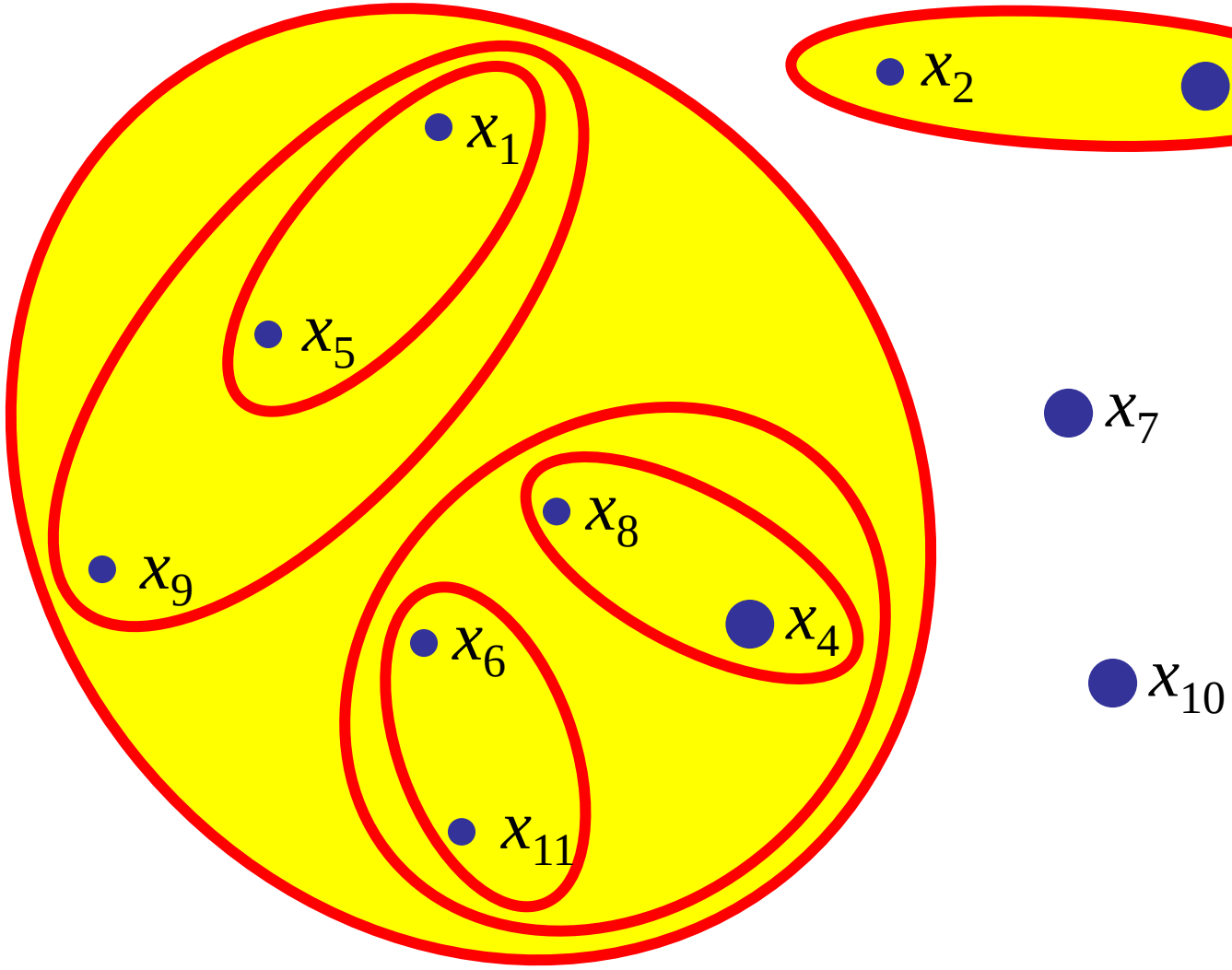
Grundlæggende Algoritmer og Datastrukturer

Union-Find [CLRS, kapitel 21.2-21.3]

Union-Find

MakeSet(x)	Opret en ny mængde $S = \{ x \}$
Union(x, y)	Erstat $S_x = \{ ..., x, ... \}$ $S_y = \{ ..., y, ... \}$ med $S_x \cup S_y = \{ ..., x, ..., y, ... \}$
FindSet(x)	Retuner en ***repræsentant*** for $S_x = \{ ..., x, ... \}$ FindSet(x) = FindSet(y) hvis og kun hvis x og y er i samme mængde

Eksempel : Union-Find



x_i	FindSet(x_i)
x_1	x_4
x_2	x_3
x_3	x_3
x_4	x_4
x_5	x_4
x_6	x_4
x_7	x_7
x_8	x_4
x_9	x_4
x_{10}	x_{10}
x_{11}	x_4

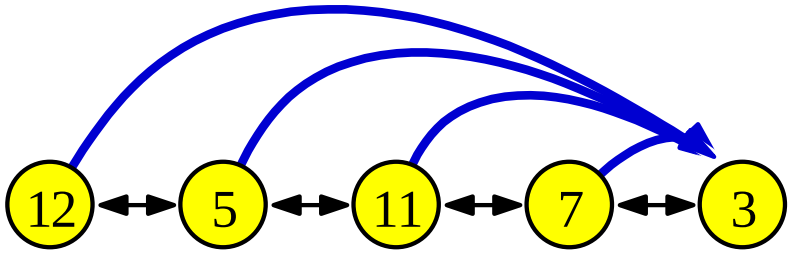
Trærepræsentation (I)

- Mængde = dobbeltkædet liste
- MakeSet = lav en ny knude
- FindSet = retuner sidste knude
- Union = konkatener listerne

MakeSet(S, x)	$O(1)$
Union(x, y)	$O(\|S_x\|+\|S_y\|)$
FindSet(x)	$O(\|S_x\|)$

Trærepræsentation (II)

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



MakeSet(S, x)	$O(1)$
Union(x, y)	$O(\min(\|S_x\|,\|S_y\|))$
FindSet(x)	$O(1)$

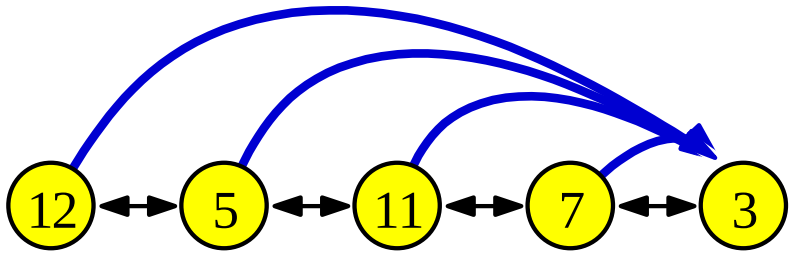
Trærepræsentation (II)
Sekvens af Union

Sætning

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

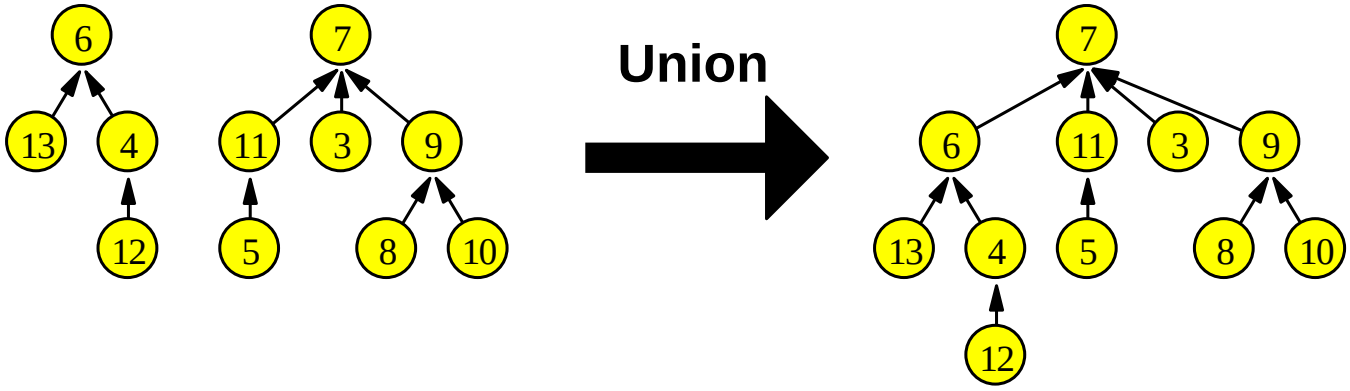
Bevis

Hver pointer flyttes højest log n gange - hver gang til en liste der mindst er dobbelt så stor



Trærepræsentation (III)

- Mængde = træ
- MakeSet = lav en ny knude
- FindSet = retuner roden
- Union = Sæt det ”lille” træ under roden af det ”store” træ (størrelse = antal elementer i træet)



Trærepræsentation (III)

Et træ der repræsenterer en mængde med *n* elementer har højde ?

a) O(1)
b) O(log n)
c) O(√n)
d) O(n)
e) Ved ikke

Stikomprimering

```
MAKE-SET(x)
1  x.p = x
2  x.rank = 0

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

LINK(x, y)
1  if x.rank > y.rank
2       y.p = x
3  else x.p = y
4       if x.rank == y.rank
5            y.rank = y.rank + 1

FIND-SET(x)
1  if x ≠ x.p
2       x.p = FIND-SET(x.p)
3  return x.p
```

linking ved rank

stikomprimering

Analyse af Trærepræsentation med Rank-Linkning

Lemma

$$height[\text{rod}] \leq rank[\text{rod}]$$
$$size[\text{rod}] \geq 2^{rank[\text{rod}]}$$

Bevis

Induktion.

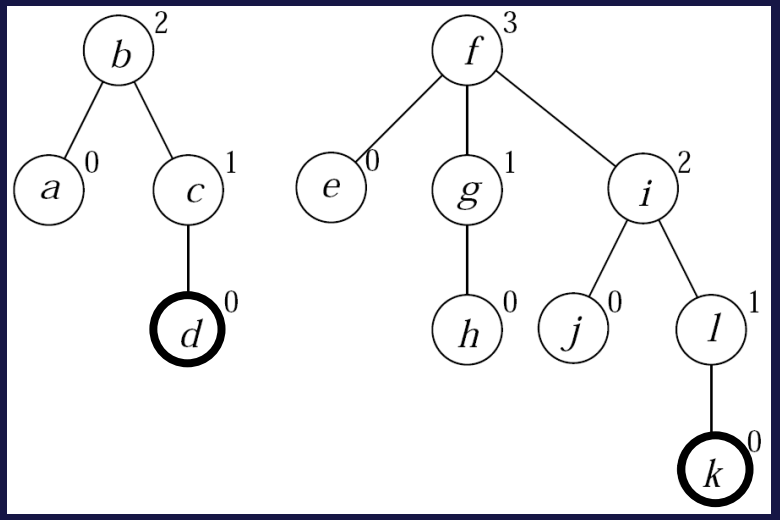
MakeSet(S**,** x**)**	$O(1)$
Union(x**,** y**)**	$O((\log \lvert S_x \rvert)+(\log \lvert S_y \rvert))$
FindSet(x**)**	$O(\log \lvert S_x \rvert)$

(Ovenstående udnytter kun linking by rank)

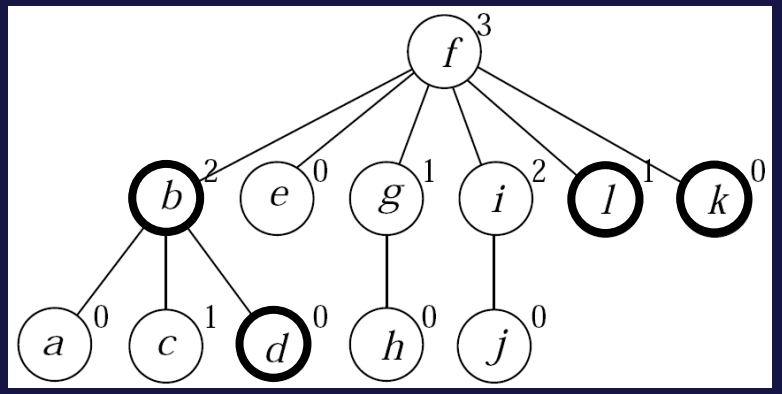
Union(d,k)

Hvor mange børn har roden efter Union-operationen er udført, når der anvendes union-by-rank og sti-komprimering ?

a) 1
b) 2
c) 3
d) 4
e) 5
f) 6
g) 7
h) 8
i) ved ikke



Eksamensopgave sommeren 2009



Analyse af Trærepræsentation med Stikomprimering

Sætning

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