

# Forelesning 5

## Bonusmateriale

**Ting som ikke ble med i forelesningen,  
men som kanskje kan være av interesse**

# Søketrær > **Minimum**

Boka kaller denne Tree-Minimum.

Å finne maksimum er helt ekvivalent/symmetrisk.

MINIMUM( $x$ )

Finn noden med minst nøkkel i deltreet med rot  $x$

```
MINIMUM( $x$ )  
1  while  $x.left \neq \text{NIL}$ 
```

Har  $x$  et venstre deltre? Da må svaret ligge der

```
MINIMUM( $x$ )  
1  while  $x.left \neq \text{NIL}$   
2       $x = x.left$ 
```

Har ikke  $x$  noe venstre deltre? Da er  $x$  minst!

```
MINIMUM( $x$ )  
1  while  $x.left \neq \text{NIL}$   
2       $x = x.left$   
3  return  $x$ 
```

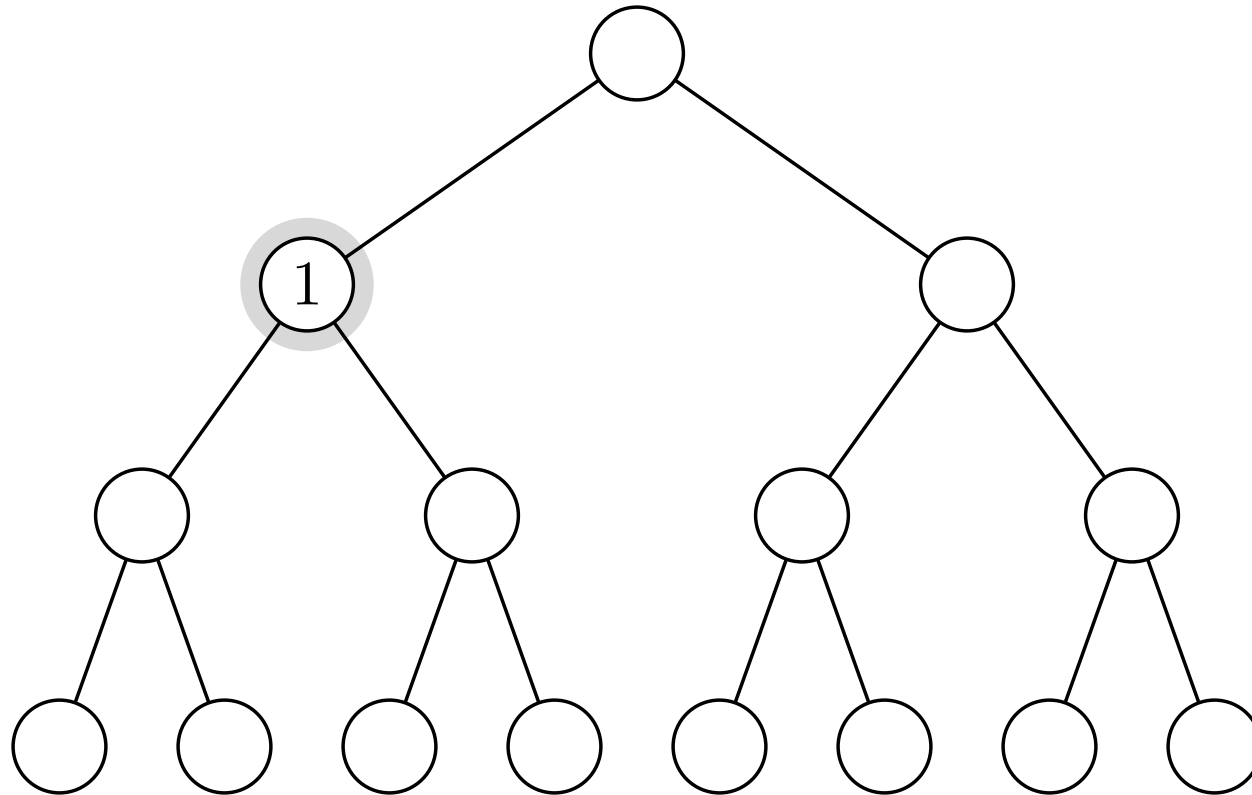
Merk: Vi har alltid  $\text{MINIMUM}(x).left == \text{NIL}$

# Søketrær > Etterfølger

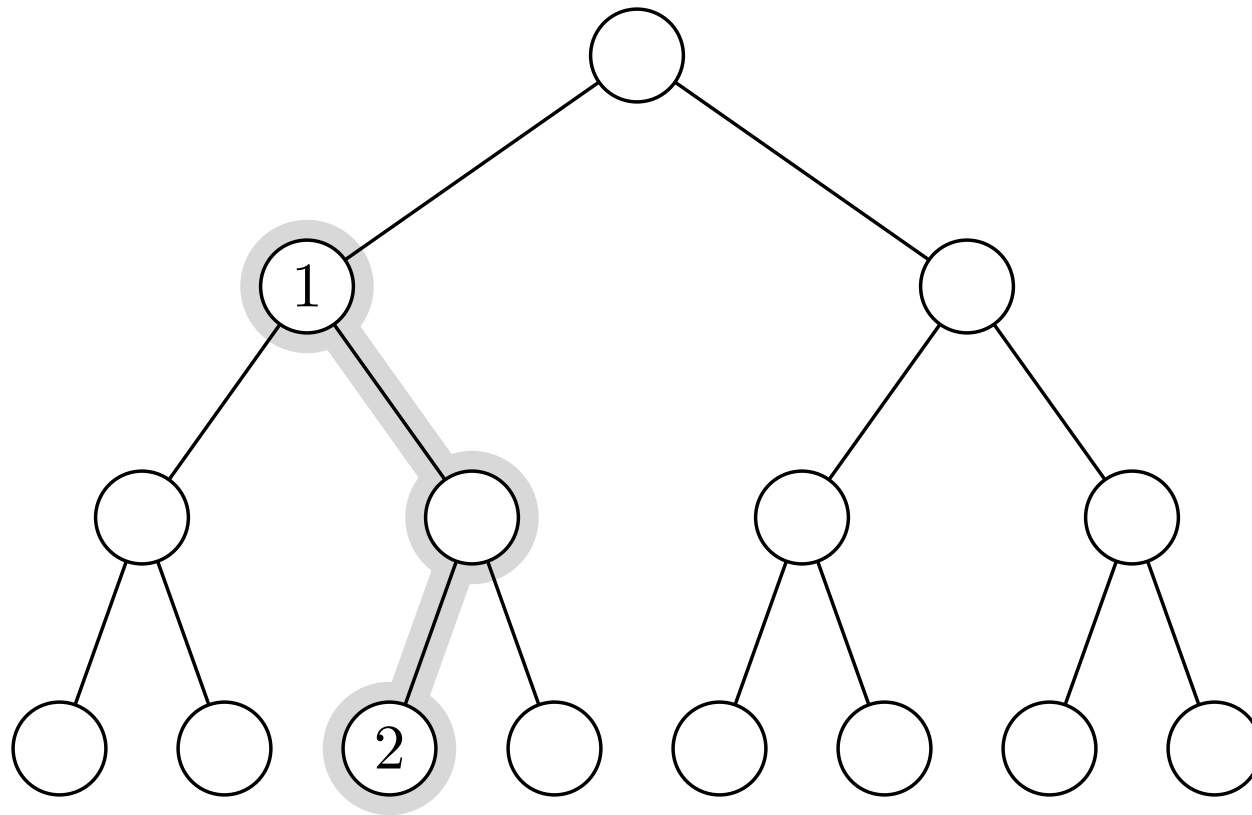
Boka kaller denne Tree-Successor.

Å finne forgjengeren i den ordnede rekkefølgen er helt ekvivalent/symmetrisk.

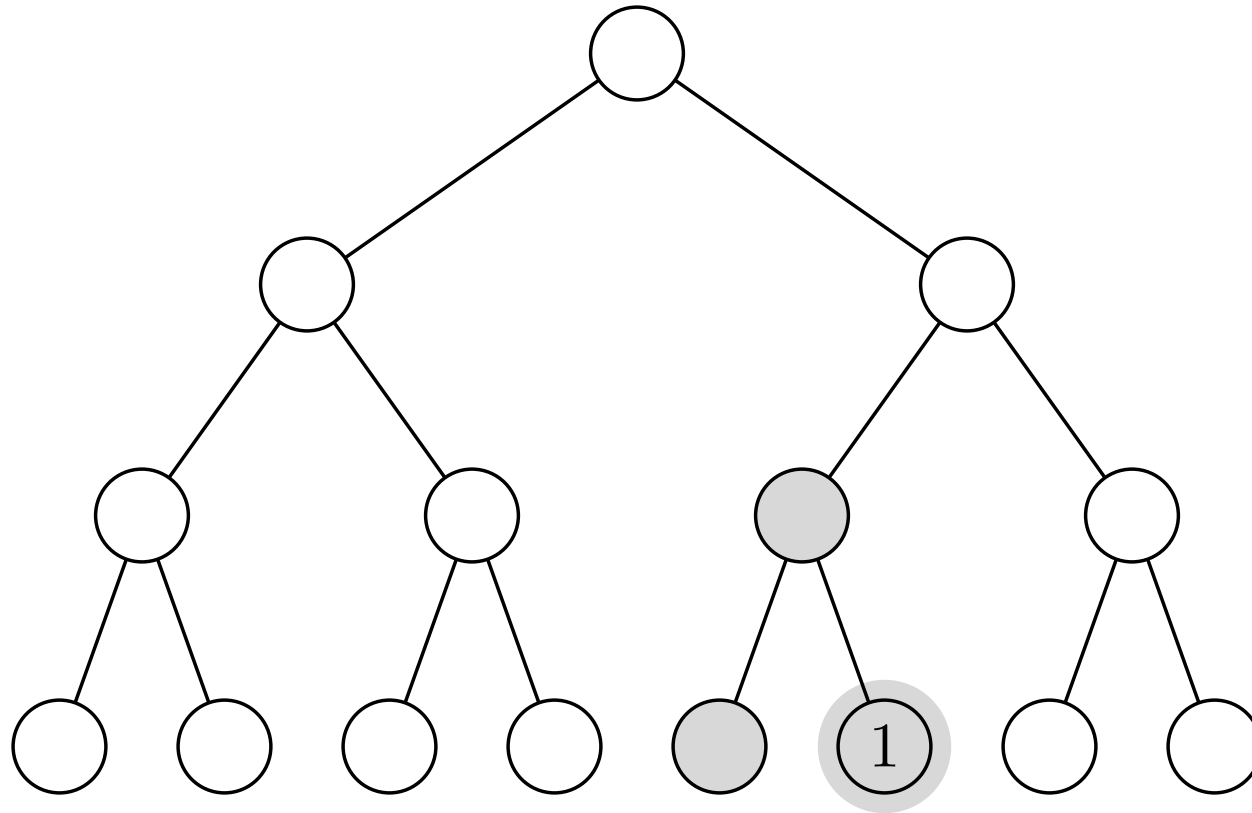




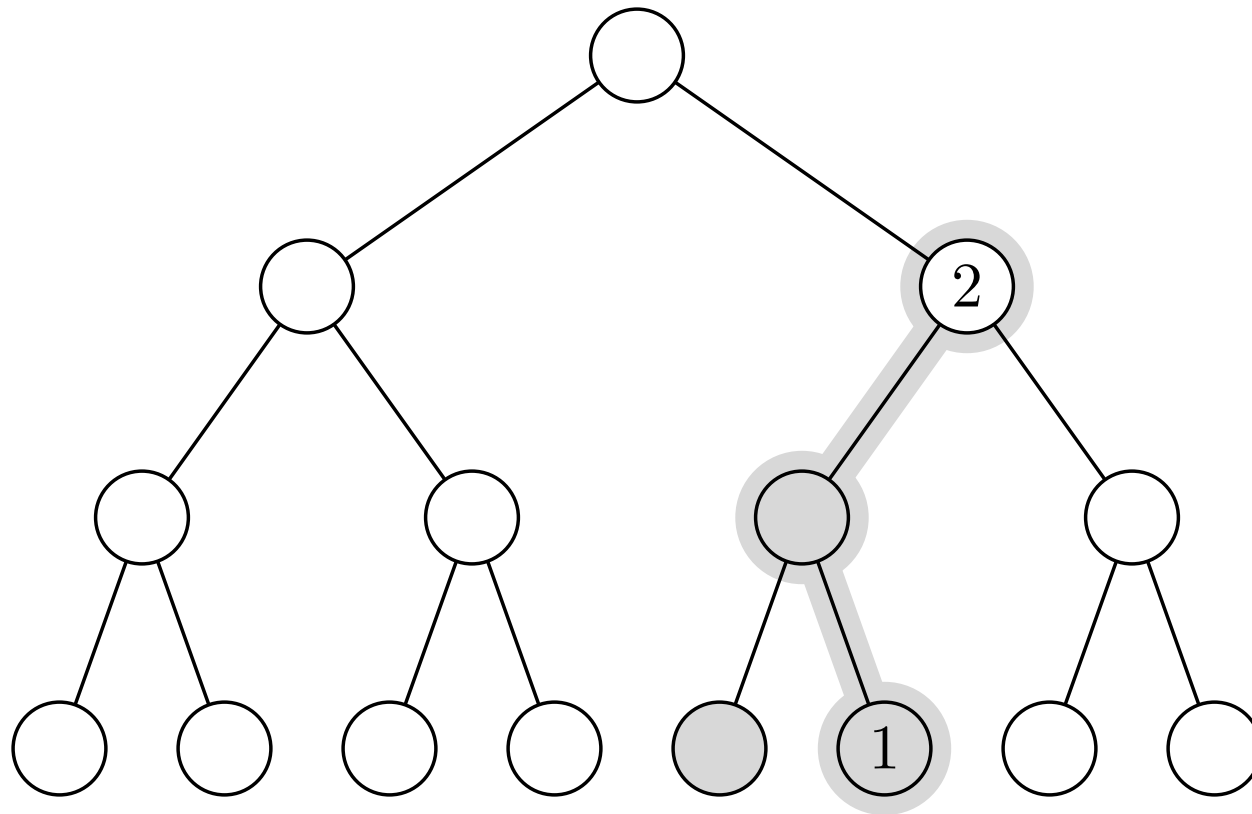
Har vi barn? Etterfølger er minimum i høyre deltre



Har vi barn? Etterfølger er minimum i høyre deltre



Ellers: Vi er maksimum i et deltre. Finn treets forelder!



Ellers: Vi er maksimum i et deltre. Finn treets forelder!

SUCCESSOR( $x$ )

Etterfølger: Finn neste node i sortert rekkefølge

SUCCESSOR( $x$ )  
1 **if**  $x.right \neq \text{NIL}$

Har  $x$  et høyre deltre? Da må svaret ligge der

```
SUCCESSOR(x)  
1  if x.right ≠ NIL  
2      return MINIMUM(x.right)
```

Har  $x$  et høyre deltre? Da må svaret ligge der

```
SUCCESSOR( $x$ )  
1  if  $x.right \neq \text{NIL}$   
2      return MINIMUM( $x.right$ )  
3  else
```

$x$  er maksimum i et deltre; finn rota ett hakk over



```
SUCCESSOR( $x$ )  
1  if  $x.right \neq \text{NIL}$   
2      return MINIMUM( $x.right$ )  
3  else  
4       $y = x.p$ 
```

$x$  er maksimum i et deltre; finn rota ett hakk over

```
SUCCESSOR( $x$ )  
1  if  $x.right \neq \text{NIL}$   
2      return MINIMUM( $x.right$ )  
3  else  
4       $y = x.p$   
5      while  $y \neq \text{NIL}$  and  $x == y.right$ 
```

$x$  er maksimum i et deltre; finn rota ett hakk over

SUCCESSOR( $x$ )

```
1  if  $x.right \neq \text{NIL}$ 
2      return MINIMUM( $x.right$ )
3  else
4       $y = x.p$ 
5      while  $y \neq \text{NIL}$  and  $x == y.right$ 
6           $x = y$ 
```

$x$  er maksimum i et deltre; finn rota ett hakk over

```
SUCCESSOR( $x$ )
1  if  $x.right \neq \text{NIL}$ 
2      return MINIMUM( $x.right$ )
3  else
4       $y = x.p$ 
5      while  $y \neq \text{NIL}$  and  $x == y.right$ 
6           $x = y$ 
7           $y = y.p$ 
```

$x$  er maksimum i et deltre; finn rota ett hakk over

SUCCESSOR( $x$ )

```
1  if  $x.right \neq \text{NIL}$ 
2      return MINIMUM( $x.right$ )
3  else
4       $y = x.p$ 
5      while  $y \neq \text{NIL}$  and  $x == y.right$ 
6           $x = y$ 
7           $y = y.p$ 
8      return  $y$ 
```

# Søketrær > Sletting

Grundig forståelse kreves ikke

Boka kaller denne Tree-Delete.

TRANSPLANT( $T, u, v$ )

Erstatt noden  $u$  med noden  $v$  i treet  $T$

```
TRANSPLANT( $T, u, v$ )  
1  if  $u.p == \text{NIL}$ 
```

Hvis  $u$  var rota til  $T$  ...



TRANSPLANT( $T, u, v$ )

1 **if**  $u.p == \text{NIL}$

2        $T.root = v$

... så blir  $v$  den nye rota

```
TRANSPLANT(T, u, v)  
1  if u.p == NIL  
2      T.root = v  
3  elseif u == u.p.left
```

Ellers: *v* blir venstre/høyre barn av *u* sin forelder

TRANSPLANT( $T, u, v$ )

1 **if**  $u.p == \text{NIL}$

2      $T.root = v$

3 **elseif**  $u == u.p.left$

4      $u.p.left = v$

TRANSPLANT( $T, u, v$ )

```
1 if  $u.p == \text{NIL}$ 
2      $T.root = v$ 
3 elseif  $u == u.p.left$ 
4      $u.p.left = v$ 
5 else  $u.p.right = v$ 
```

```
TRANSPLANT( $T, u, v$ )  
1  if  $u.p == \text{NIL}$   
2       $T.root = v$   
3  elseif  $u == u.p.left$   
4       $u.p.left = v$   
5  else  $u.p.right = v$   
6  if  $v \neq \text{NIL}$ 
```

Oppdater  $v$  sin foreldrepeker, om nødvendig/mulig

```
TRANSPLANT( $T, u, v$ )  
1  if  $u.p == \text{NIL}$   
2       $T.root = v$   
3  elseif  $u == u.p.left$   
4       $u.p.left = v$   
5  else  $u.p.right = v$   
6  if  $v \neq \text{NIL}$   
7       $v.p = u.p$ 
```

**Transplantering fikser bare ting som har med foreldrenoden å gjøre. Under sletting må vi også håndtere barna.**

DELETE( $T, z$ )

Slett noden  $z$  fra treet  $T$



```
DELETE(T, z)
  1  if z.left == NIL
```

Bare (dvs., maks) ett barn? Bare erstatt  $z$  med barnet

```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
```

Bare (dvs., maks) ett barn? Bare erstatt  $z$  med barnet

```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
3  elseif z.right == NIL
```

Bare (dvs., maks) ett barn? Bare erstatt  $z$  med barnet

```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
3  elseif z.right == NIL
4      TRANSP(T, z, z.left)
```

Bare (dvs., maks) ett barn? Bare erstatt  $z$  med barnet

```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
3  elseif z.right == NIL
4      TRANSP(T, z, z.left)
5  else y = MINIMUM(z.right)
```

Har to barn. Finn etterfølger *y* (som i SUCCESSOR)

```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
3  elseif z.right == NIL
4      TRANSP(T, z, z.left)
5  else y = MINIMUM(z.right)
6      if y ≠ z.right
```

Vil erstatte  $z$  med  $y$ ; her kreves litt fiksing

```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
3  elseif z.right == NIL
4      TRANSP(T, z, z.left)
5  else y = MINIMUM(z.right)
6      if y ≠ z.right
7          TRANSP(T, y, y.right)
```

Kan ikke ha noe venstre barn; kan dermed erstattes med høyre barn

DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
```

La  $y$  ta over  $z$  sitt høyre barn



DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
```

La  $y$  ta over  $z$  sitt høyre barn

```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
3  elseif z.right == NIL
4      TRANSP(T, z, z.left)
5  else y = MINIMUM(z.right)
6      if y ≠ z.right
7          TRANSP(T, y, y.right)
8          y.right = z.right
9          y.right.p = y
10     TRANSP(T, z, y)
```

Enten er  $y$  høyre barn, ellers har den tatt over høyre barn

```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
3  elseif z.right == NIL
4      TRANSP(T, z, z.left)
5  else y = MINIMUM(z.right)
6      if y ≠ z.right
7          TRANSP(T, y, y.right)
8          y.right = z.right
9          y.right.p = y
10     TRANSP(T, z, y)
11     y.left = z.left
```

Etter at vi har erstattet *z*: Ta over venstre barn også

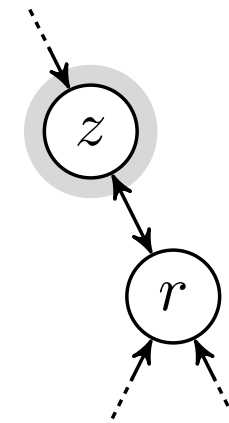
```
DELETE(T, z)
1  if z.left == NIL
2      TRANSP(T, z, z.right)
3  elseif z.right == NIL
4      TRANSP(T, z, z.left)
5  else y = MINIMUM(z.right)
6      if y ≠ z.right
7          TRANSP(T, y, y.right)
8          y.right = z.right
9          y.right.p = y
10     TRANSP(T, z, y)
11     y.left = z.left
12     y.left.p = y
```

Etter at vi har erstattet *z*: Ta over venstre barn også

Søketrær › Sletting › **Uten v. barn**

DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```

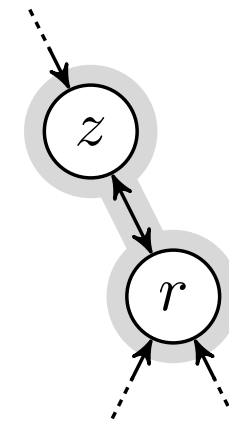


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```



DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```

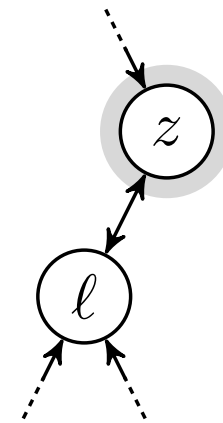




Søketrær › Sletting › **Uten h. barn**

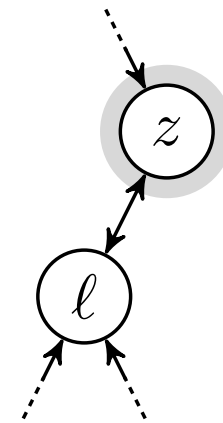
DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```



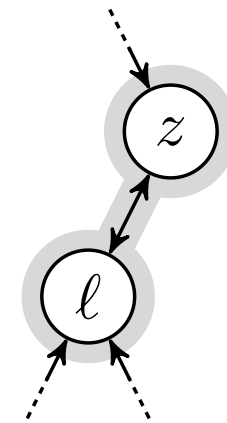
DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```



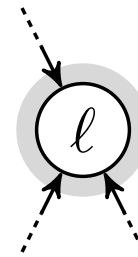
DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```



DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```



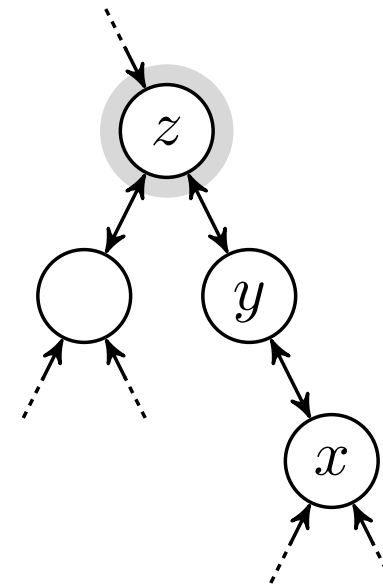
Søketrær > Sletting > **y er barn**

DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```

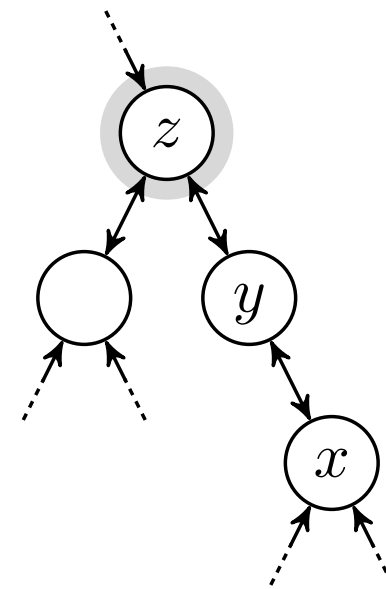


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

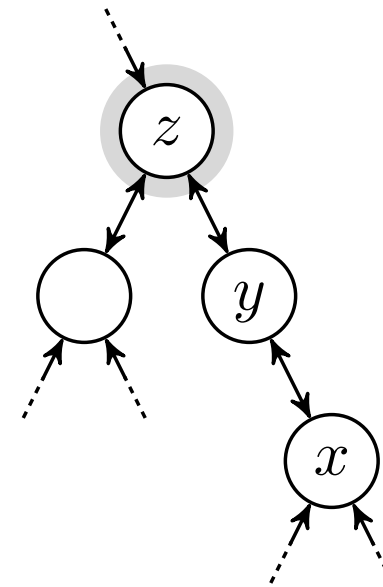
```





DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```

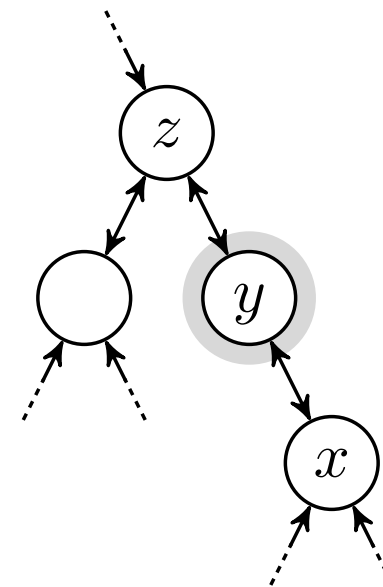


DELETE( $T, z$ )

```

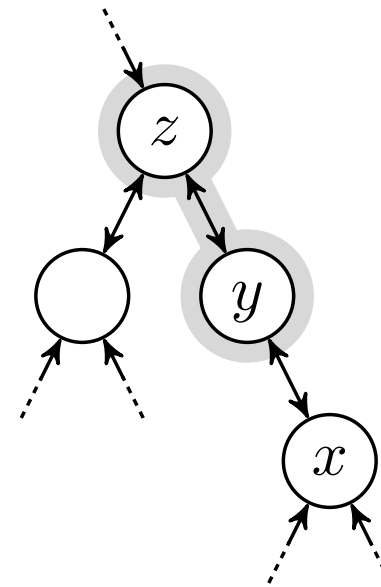
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6  if  $y \neq z.right$ 
7      TRANSP( $T, y, y.right$ )
8       $y.right = z.right$ 
9       $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```



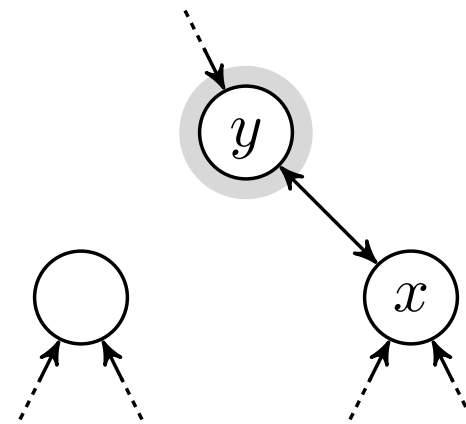
DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```



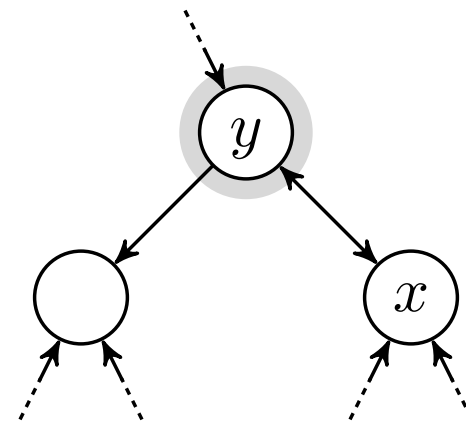
DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```



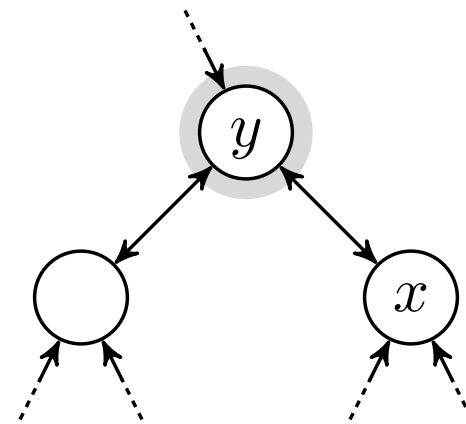
DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```



DELETE( $T, z$ )

```
1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 
```



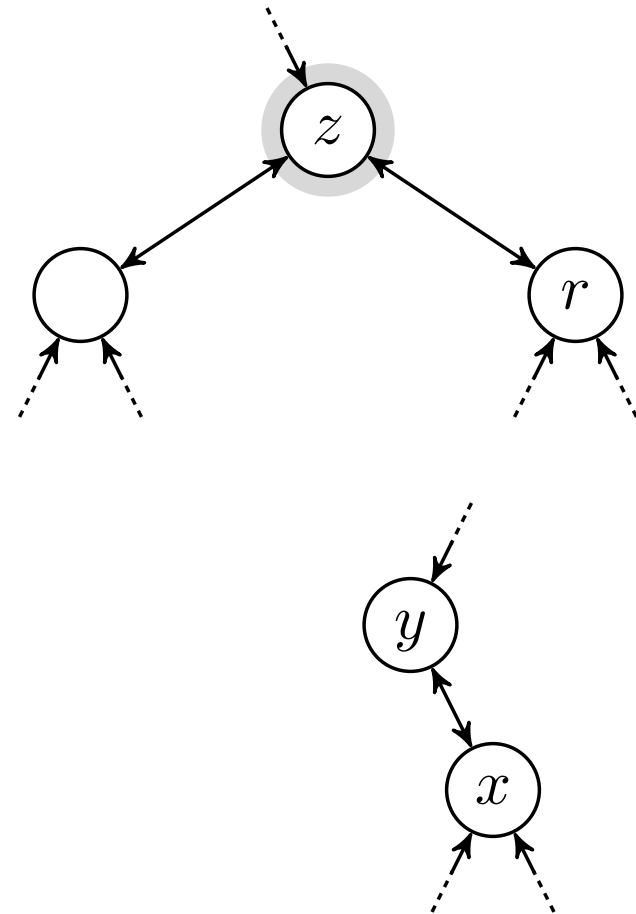
Søketrær › Sletting › **y er ikke barn**

DELETE( $T, z$ )

```

1  if  $z.left == NIL$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == NIL$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = MINIMUM(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```



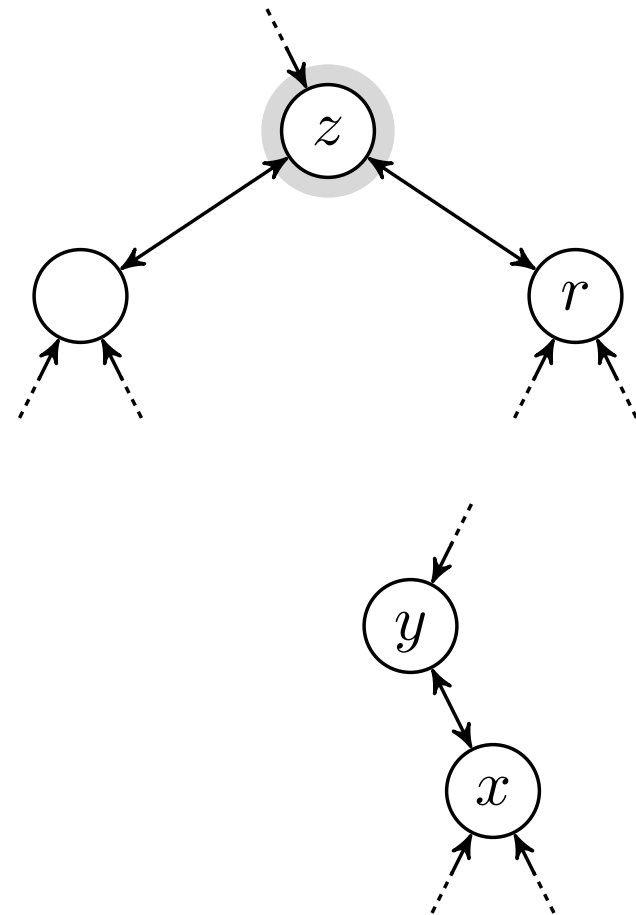


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```

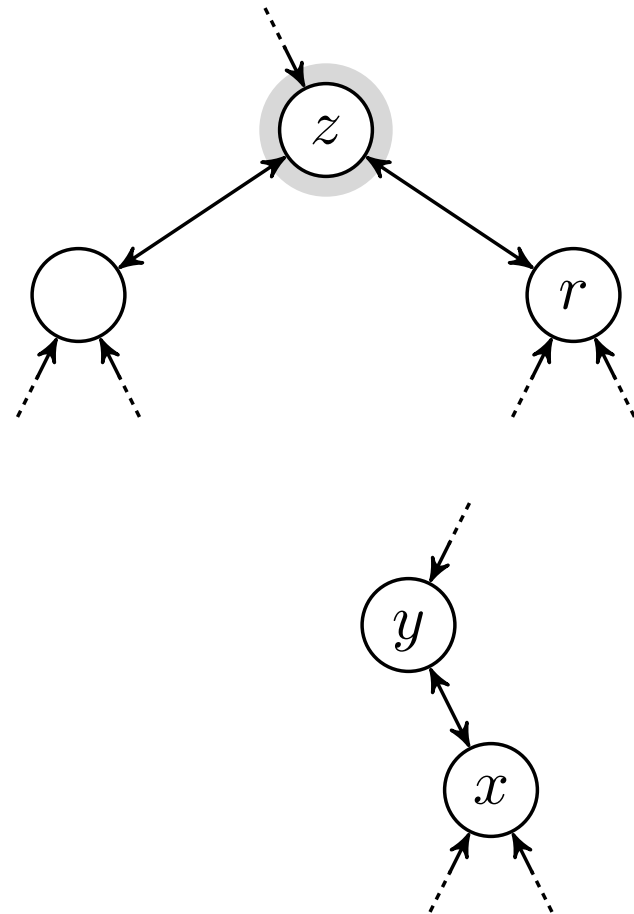


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

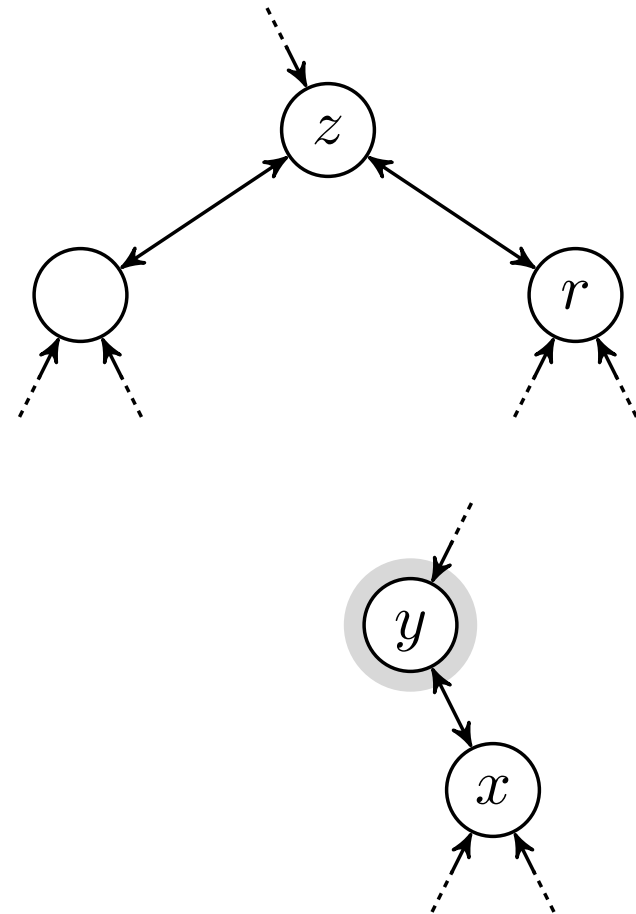
```



DELETE( $T, z$ )

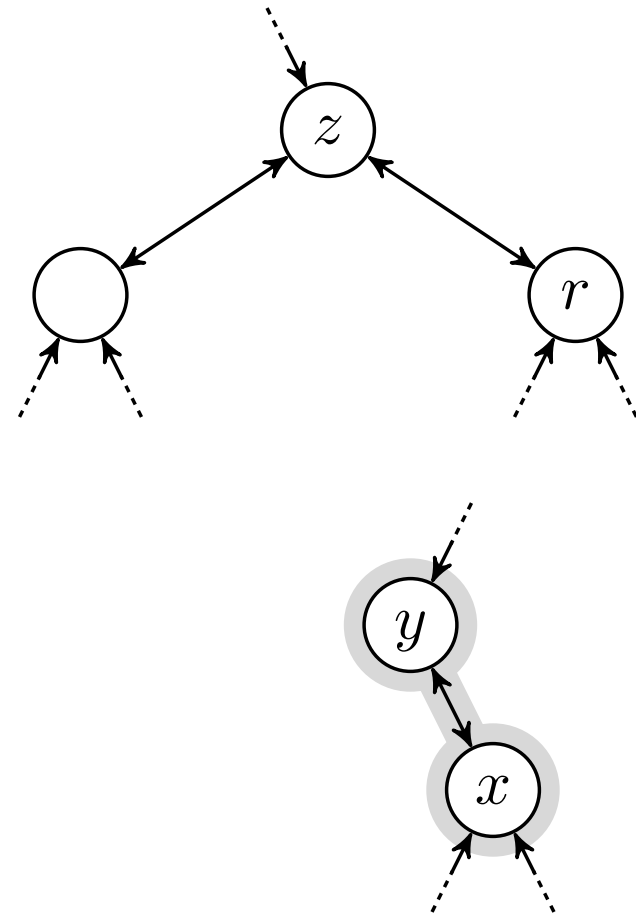
```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6  if  $y \neq z.right$ 
7      TRANSP( $T, y, y.right$ )
8       $y.right = z.right$ 
9       $y.right.p = y$ 
10 TRANSP( $T, z, y$ )
11  $y.left = z.left$ 
12  $y.left.p = y$ 
    
```



DELETE( $T, z$ )

- 1 **if**  $z.left == \text{NIL}$
- 2     TRANSP( $T, z, z.right$ )
- 3 **elseif**  $z.right == \text{NIL}$
- 4     TRANSP( $T, z, z.left$ )
- 5 **else**  $y = \text{MINIMUM}(z.right)$
- 6     **if**  $y \neq z.right$
- 7         TRANSP( $T, y, y.right$ )
- 8          $y.right = z.right$
- 9          $y.right.p = y$
- 10     TRANSP( $T, z, y$ )
- 11      $y.left = z.left$
- 12      $y.left.p = y$

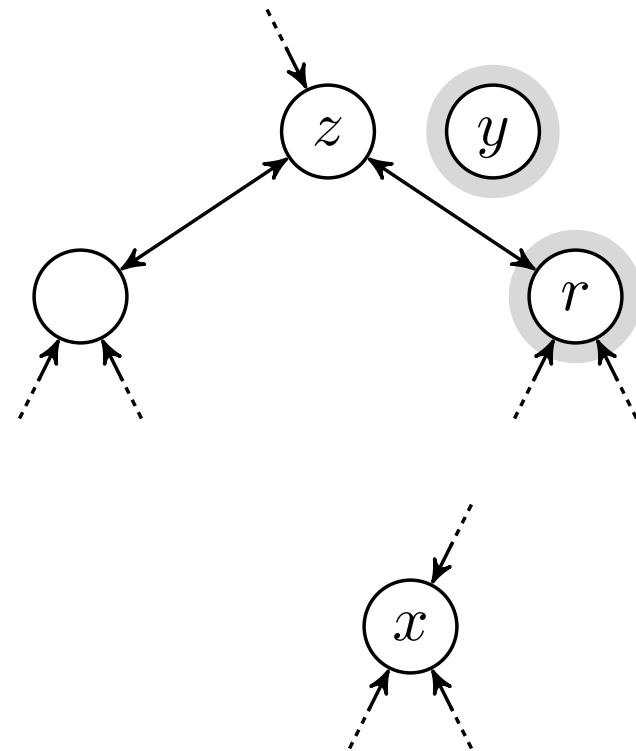


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```

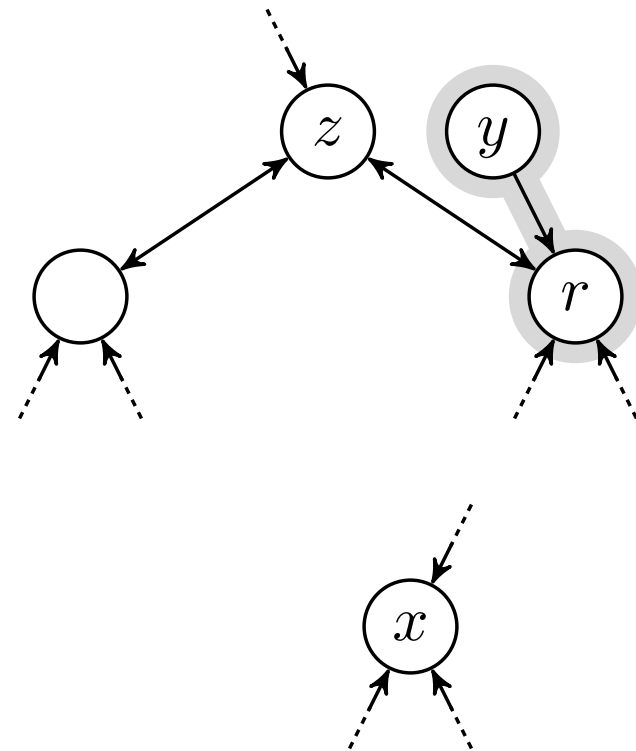


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```

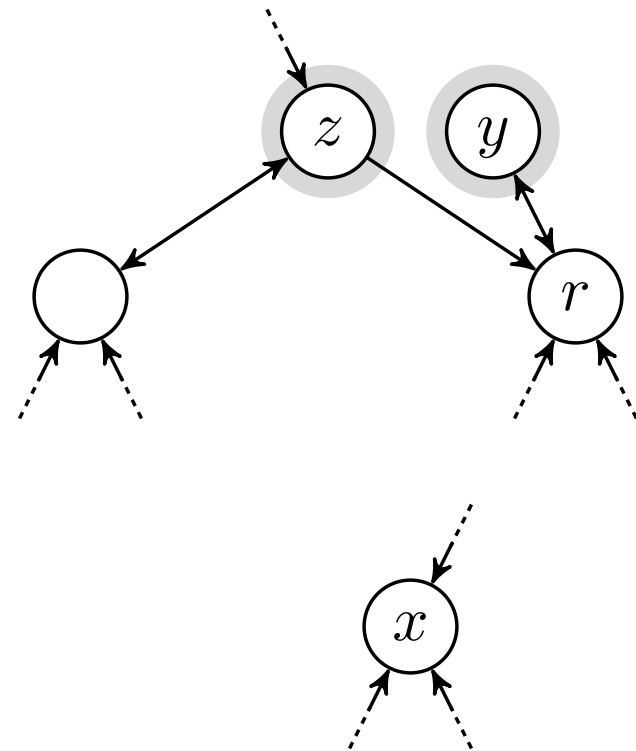


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```

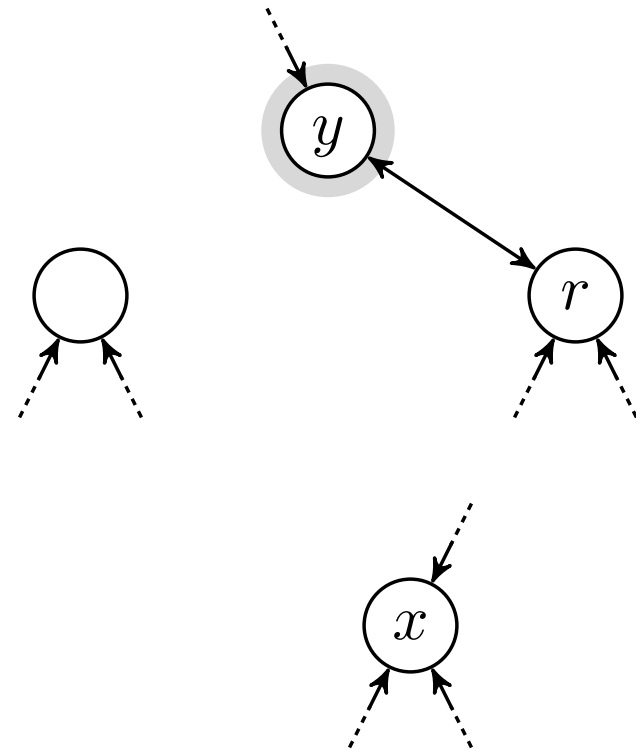


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```



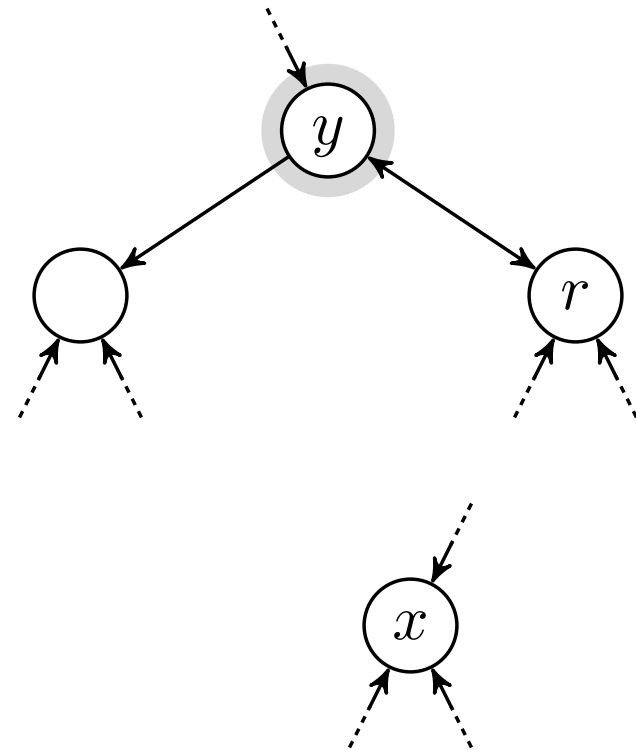


DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```



DELETE( $T, z$ )

```

1  if  $z.left == \text{NIL}$ 
2      TRANSP( $T, z, z.right$ )
3  elseif  $z.right == \text{NIL}$ 
4      TRANSP( $T, z, z.left$ )
5  else  $y = \text{MINIMUM}(z.right)$ 
6      if  $y \neq z.right$ 
7          TRANSP( $T, y, y.right$ )
8           $y.right = z.right$ 
9           $y.right.p = y$ 
10     TRANSP( $T, z, y$ )
11      $y.left = z.left$ 
12      $y.left.p = y$ 

```

