

Forelesning 3

Bonusmateriale

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

Designskjema

Instans	Løsning
Dekomponering	Kombinasjon IS
Delinstans	Delløsninger IH
Grunntilfelle	Grunnløsning

Instans	Løsning
Dekomponering	Kombinasjon IS
Delinstans	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva er input?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning
Dekomponering	Kombinasjon IS
Delinstans	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva er input?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning
Dekomponering	Kombinasjon IS
Delinstans	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva ønsker vi at output skal være?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering	Kombinasjon IS
Delinstans	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva ønsker vi at output skal være?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering	Kombinasjon IS
Delinstans	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvordan kan vi krympe eller dele opp instansen?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A?	Kombinasjon IS
Delinstans	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvordan kan vi krympe eller dele opp instansen?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans	Delløsninger IH
Grunttilfelle	Grunnløsning

Hva har vi brutt problemet ned til?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva har vi brutt problemet ned til?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvor stopper dekomponeringen?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans Venstre eller høyre halvdel, $A[p:r]$...	Delløsninger IH
Grunntilfelle Intervall av lengde 1, altså der $p == r$	Grunnløsning

Hvor stopper dekomponeringen?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delloøsninger IH
Grunntilfelle Intervall av lengde 1, altså der $p == r$	Grunnløsning

Hva er løsningen da?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delløsninger IH
Grunntilfelle Intervall av lengde 1, altså der $p == r$	Grunnløsning Hvis $A[p] = v$ er løsningen p ; ellers NIL

Hva er løsningen da?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delløsninger IH
Grunntilfelle Intervall av lengde 1, altså der $p == r$	Grunnløsning Hvis $A[p] = v$ er løsningen p ; ellers NIL


Hva er løsningene for delinstansene?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delløsninger IH Indeks fra riktig halvdel, eller NIL
Grunntilfelle Intervall av lengde 1, altså der $p == r$	Grunnløsning Hvis $A[p] = v$ er løsningen p ; ellers NIL


Hva er løsningene for delinstansene?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delløsninger IH Indeks fra riktig halvdel, eller NIL
Grunntilfelle Intervall av lengde 1, altså der $p == r$	Grunnløsning Hvis $A[p] = v$ er løsningen p ; ellers NIL

Hvordan bygger vi en full løsning av delløsninger?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS 
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delløsninger IH Indeks fra riktig halvdel, eller NIL
Grunntilfelle Intervall av lengde 1, altså der $p == r$	Grunnløsning Hvis $A[p] = v$ er løsningen p ; ellers NIL

Hvordan bygger vi en full løsning av delløsninger?

Instans En ikke tom, sortert sekvens A og en verdi v	Løsning Om mulig, en indeks q , så $A[q] = v$; ellers NIL
Dekomponering Er v i venstre eller høyre halvdel av A ?	Kombinasjon IS 
Delinstans Venstre eller høyre halvdel, $A[p:r]$	Delløsninger IH Indeks fra riktig halvdel, eller NIL
Grunntilfelle Intervall av lengde 1, altså der $p == r$	Grunnløsning Hvis $A[p] = v$ er løsningen p ; ellers NIL

Instans	Løsning
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Instans	Løsning
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva er input?

Instans En sekvens A	Løsning
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva er input?

Instans	Løsning
En sekvens A	
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva ønsker vi at output skal være?

Instans	Løsning
En sekvens A	A, sortert
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva ønsker vi at output skal være?

Instans En sekvens A	Løsning A, sortert
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvordan kan vi krympe eller dele opp instansen?

Instans En sekvens A	Løsning A, sortert
Dekomponering Del på midten	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvordan kan vi krympe eller dele opp instansen?

Instans En sekvens A	Løsning A, sortert
Dekomponering Del på midten	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva har vi brutt problemet ned til?

Instans En sekvens A	Løsning A, sortert
Dekomponering Del på midten	Kombinasjon IS
Delinstanser Venstre og høyre halvdel	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva har vi brutt problemet ned til?

Instans En sekvens A	Løsning A, sortert
Dekomponering Del på midten	Kombinasjon IS
Delinstanser Venstre og høyre halvdel	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvor stopper dekomponeringen?

Instans En sekvens A	Løsning A, sortert
Dekomponering Del på midten	Kombinasjon IS
Delinstanser Venstre og høyre halvdel	Delløsninger IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning

Hvor stopper dekomponeringen?

Instans En sekvens A	Løsning A, sortert
Dekomponering Del på midten	Kombinasjon IS
Delinstanser Venstre og høyre halvdel	Delløsninger IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning

Hva er løsningen da?

Instans En sekvens A	Løsning A, sortert
Dekomponering Del på midten	Kombinasjon IS
Delinstanser Venstre og høyre halvdel	Delløsninger IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert

Hva er løsningen da?

Instans En sekvens A	Løsning A, sortert
Dekomponering Del på midten	Kombinasjon IS
Delinstanser Venstre og høyre halvdel	Delløsninger IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert

Hva er løsningene for delinstansene?

Instans En sekvens A	Løsning A, sortert	
Dekomponering Del på midten	Kombinasjon	IS
Delinstanser Venstre og høyre halvdel	Delløsninger Venstre og høyre halvdel, sorterte	IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert	

Hva er løsningene for delinstansene?

Instans En sekvens A	Løsning A, sortert	
Dekomponering Del på midten	Kombinasjon	IS
Delinstanser Venstre og høyre halvdel	Delløsninger Venstre og høyre halvdel, sorterte	IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert	

Hvordan bygger vi en full løsning av delløsninger?

Instans En sekvens A	Løsning A, sortert	
Dekomponering Del på midten	Kombinasjon Flett venstre og høyre halvdel	IS
Delinstanser Venstre og høyre halvdel	Delløsninger Venstre og høyre halvdel, sorterte	IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert	

Hvordan bygger vi en full løsning av delløsninger?

Instans En sekvens A	Løsning A, sortert	
Dekomponering Del på midten	Kombinasjon Flett venstre og høyre halvdel	IS
Delinstanser Venstre og høyre halvdel	Delløsninger Venstre og høyre halvdel, sorterte	IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert	

Instans	Løsning
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Instans	Løsning
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva er input?

Instans En sekvens A	Løsning
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva er input?

Instans En sekvens A	Løsning
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva ønsker vi at output skal være?

Instans	Løsning
En sekvens A	A, sortert
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva ønsker vi at output skal være?

Instans En sekvens A	Løsning A, sortert
Dekomponering	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvordan kan vi krympe eller dele opp instansen?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvordan kan vi krympe eller dele opp instansen?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva har vi brutt problemet ned til?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger IH
Grunntilfelle	Grunnløsning

Hva har vi brutt problemet ned til?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger IH
Grunntilfelle	Grunnløsning

Hvor stopper dekomponeringen?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning

Hvor stopper dekomponeringen?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning

Hva er løsningen da?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert

Hva er løsningen da?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert

Hva er løsningene for delinstansene?

Instans En sekvens A	Løsning A, sortert
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger IH Liten og stor del, til venstre og høyre, sorterte
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert

Hva er løsningene for delinstansene?

Instans En sekvens A	Løsning A, sortert	
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon	IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger Liten og stor del, til venstre og høyre, sorterte	IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert	

Hvordan bygger vi en full løsning av delløsninger?

Instans En sekvens A	Løsning A, sortert		
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon	↑	IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger		IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning		

Hvordan bygger vi en full løsning av delløsninger?

Instans En sekvens A	Løsning A, sortert		
Dekomponering Fordel i to deler, venstre verdier mindre enn høyre	Kombinasjon	↑	IS
Delinstanser Liten og stor del, til venstre og høyre	Delløsninger		IH
Grunntilfelle Sekvens med maks ett element	Grunnløsning Samme sekvens, allerede sortert		

Variabelskifte

$$T(\sqrt{n}) = \lg n$$

$$T(\sqrt{n}) = \lg n$$

$$T(n^{\frac{1}{2}}) = \lg n$$

$$m \stackrel{\text{def}}{=} \lg n$$

For å bli kvitt logaritmen: Gi $\lg n$ et nytt navn

D&C › variabelskifte › $T(n^{\frac{1}{2}}) = \lg n$

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^{\frac{m}{2}}) = m$$

Potensen er fortsatt problematisk

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^{\frac{m}{2}}) = m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

Gi $T(2^m)$ et nytt navn!

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^{\frac{m}{2}}) = m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

$$T(2^{\frac{m}{2}}) = S(m/2)$$

Hvis $S(m) = T(2^m)$ så må $S(m/2)$ være $T(2^{\frac{m}{2}})$

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^{\frac{m}{2}}) = m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

$$T(2^{\frac{m}{2}}) = S(m/2)$$

$$S(m/2) = m$$

Slik ser ligningen vår ut med nye navn

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^{\frac{m}{2}}) = m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

$$T(2^{\frac{m}{2}}) = S(m/2)$$

$$S(m/2) = m$$

$$S(m) = 2m$$

(Evt. enda et skifte: $x = m/2 \implies S(x) = 2x \implies S(m) = 2m$)

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^{\frac{m}{2}}) = m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

$$T(2^{\frac{m}{2}}) = S(m/2)$$

$$S(m/2) = m$$

$$S(m) = 2m$$

$$T(n) = 2 \lg n$$

Her bruker vi bare definisjonene våre: $S(m) = T(n)$ og $m = \lg n$

$$T(n) = 2T(\sqrt{n}) + \lg n$$

$$T(n) = 2T(\sqrt{n}) + \lg n$$

$$T(n) = 2T(n^{\frac{1}{2}}) + \lg n$$

D&C › variabelskifte › $T(n) = 2T(n^{\frac{1}{2}}) + \lg n$

$$m \stackrel{\text{def}}{=} \lg n$$

For å bli kvitt logaritmen: Gi $\lg n$ et nytt navn

D&C › variabelskifte › $T(n) = 2T(n^{\frac{1}{2}}) + \lg n$

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^m) = T(2^{\frac{m}{2}}) + m$$

Potensen er fortsatt problematisk

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^m) = T(2^{\frac{m}{2}}) + m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

Gi $T(2^m)$ et nytt navn!

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^m) = T(2^{\frac{m}{2}}) + m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

$$T(2^{\frac{m}{2}}) = S(m/2)$$

Hvis $S(m) = T(2^m)$ så må $S(m/2)$ være $T(2^{\frac{m}{2}})$

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^m) = T(2^{\frac{m}{2}}) + m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

$$T(2^{\frac{m}{2}}) = S(m/2)$$

$$S(m) = 2S(m/2) + m$$

Slik ser ligningen vår ut med nye navn

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^m) = T(2^{\frac{m}{2}}) + m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

$$T(2^{\frac{m}{2}}) = S(m/2)$$

$$S(m) = 2S(m/2) + m$$

$$S(m) = m \lg m + m$$

Standard rekurrensløsning, som tidligere i dag

$$m \stackrel{\text{def}}{=} \lg n$$

$$T(2^m) = T(2^{\frac{m}{2}}) + m$$

$$S(m) \stackrel{\text{def}}{=} T(n) = T(2^m)$$

$$T(2^{\frac{m}{2}}) = S(m/2)$$

$$S(m) = 2S(m/2) + m$$

$$S(m) = m \lg m + m$$

$$T(n) = \lg n \lg \lg n + \lg n$$

Her bruker vi bare definisjonene våre: $S(m) = T(n)$ og $m = \lg n$

Mer om kjøretid, Merge sort

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

TIME(A, p, r)

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

TIME(A, p, r)

1 **if** $p \geq r$

MERGE-SORT(A, p, r)

```
1 if  $p \geq r$ 
2     return
3  $q = \lfloor (p + r) / 2 \rfloor$ 
4 MERGE-SORT( $A, p, q$ )
5 MERGE-SORT( $A, q + 1, r$ )
6 MERGE( $A, p, q, r$ )
```

TIME(A, p, r)

```
1 if  $p \geq r$ 
2     return 1
```


MERGE-SORT(A, p, r)

```
1 if  $p \geq r$ 
2     return
3  $q = \lfloor (p + r) / 2 \rfloor$ 
4 MERGE-SORT( $A, p, q$ )
5 MERGE-SORT( $A, q + 1, r$ )
6 MERGE( $A, p, q, r$ )
```

TIME(A, p, r)

```
1 if  $p \geq r$ 
2     return 1
3  $q = \lfloor (p + r) / 2 \rfloor$ 
```

MERGE-SORT(A, p, r)

```
1 if  $p \geq r$ 
2     return
3  $q = \lfloor (p + r) / 2 \rfloor$ 
4 MERGE-SORT( $A, p, q$ )
5 MERGE-SORT( $A, q + 1, r$ )
6 MERGE( $A, p, q, r$ )
```

TIME(A, p, r)

```
1 if  $p \geq r$ 
2     return 1
3  $q = \lfloor (p + r) / 2 \rfloor$ 
4  $t_1 = \text{TIME}(A, p, q)$ 
```

MERGE-SORT(A, p, r)

```
1 if  $p \geq r$ 
2     return
3  $q = \lfloor (p + r) / 2 \rfloor$ 
4 MERGE-SORT( $A, p, q$ )
5 MERGE-SORT( $A, q + 1, r$ )
6 MERGE( $A, p, q, r$ )
```

TIME(A, p, r)

```
1 if  $p \geq r$ 
2     return 1
3  $q = \lfloor (p + r) / 2 \rfloor$ 
4  $t_1 = \text{TIME}(A, p, q)$ 
5  $t_2 = \text{TIME}(A, q + 1, r)$ 
```

MERGE-SORT(A, p, r)

```
1 if  $p \geq r$ 
2     return
3  $q = \lfloor (p + r) / 2 \rfloor$ 
4 MERGE-SORT( $A, p, q$ )
5 MERGE-SORT( $A, q + 1, r$ )
6 MERGE( $A, p, q, r$ )
```

TIME(A, p, r)

```
1 if  $p \geq r$ 
2     return 1
3  $q = \lfloor (p + r) / 2 \rfloor$ 
4  $t_1 = \text{TIME}(A, p, q)$ 
5  $t_2 = \text{TIME}(A, q + 1, r)$ 
6 return  $t_1 + t_2 + n$ 
```

TIME(A, p, r)

1 **if** $p \geq r$

2 **return** 1

3 $q = \lfloor (p + r)/2 \rfloor$

4 $t_1 = \text{TIME}(A, p, q)$

5 $t_2 = \text{TIME}(A, q + 1, r)$

6 **return** $t_1 + t_2 + n$

TIME(n)

TIME(A, p, r)

1 **if** $p \geq r$

2 **return** 1

3 $q = \lfloor (p + r)/2 \rfloor$

4 $t_1 = \text{TIME}(A, p, q)$

5 $t_2 = \text{TIME}(A, q + 1, r)$

6 **return** $t_1 + t_2 + n$

TIME(n)

1 **if** $n \geq 1$

TIME(A, p, r)

1 **if** $p \geq r$

2 **return** 1

3 $q = \lfloor (p + r)/2 \rfloor$

4 $t_1 = \text{TIME}(A, p, q)$

5 $t_2 = \text{TIME}(A, q + 1, r)$

6 **return** $t_1 + t_2 + n$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

TIME(A, p, r)

1 **if** $p \geq r$

2 **return** 1

3 $q = \lfloor (p + r)/2 \rfloor$

4 $t_1 = \text{TIME}(A, p, q)$

5 $t_2 = \text{TIME}(A, q + 1, r)$

6 **return** $t_1 + t_2 + n$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

TIME(A, p, r)

1 **if** $p \geq r$

2 **return** 1

3 $q = \lfloor (p + r)/2 \rfloor$

4 $t_1 = \text{TIME}(A, p, q)$

5 $t_2 = \text{TIME}(A, q + 1, r)$

6 **return** $t_1 + t_2 + n$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

TIME(A, p, r)

1 **if** $p \geq r$

2 **return** 1

3 $q = \lfloor (p + r)/2 \rfloor$

4 $t_1 = \text{TIME}(A, p, q)$

5 $t_2 = \text{TIME}(A, q + 1, r)$

6 **return** $t_1 + t_2 + n$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

TIME(A, p, r)

1 **if** $p \geq r$

2 **return** 1

3 $q = \lfloor (p + r)/2 \rfloor$

4 $t_1 = \text{TIME}(A, p, q)$

5 $t_2 = \text{TIME}(A, q + 1, r)$

6 **return** $t_1 + t_2 + n$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

TIME(A, p, r)

```
1 if  $p \geq r$ 
2     return 1
3  $q = \lfloor (p + r)/2 \rfloor$ 
4  $t_1 = \text{TIME}(A, p, q)$ 
5  $t_2 = \text{TIME}(A, q + 1, r)$ 
6 return  $t_1 + t_2 + n$ 
```

TIME(n)

```
1 if  $n \geq 1$ 
2     return 1
3  $m = \lfloor n/2 \rfloor$ 
4  $t_1 = \text{TIME}(m)$ 
5  $t_2 = \text{TIME}(n - m)$ 
6 return  $t_1 + t_2 + n$ 
```

$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

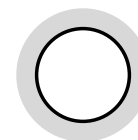
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

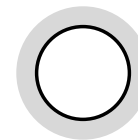
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

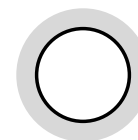
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

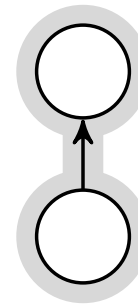
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

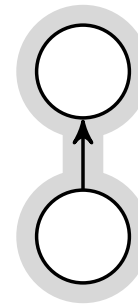
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

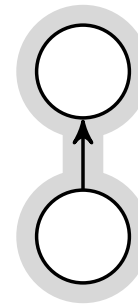
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

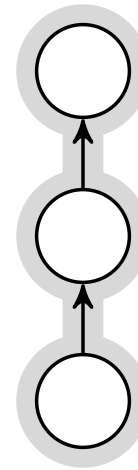
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

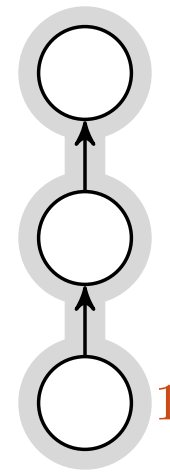
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

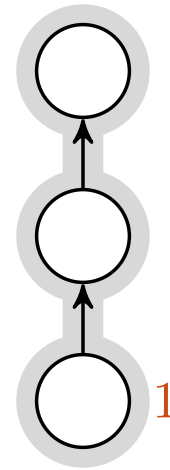
4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

→ 1

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

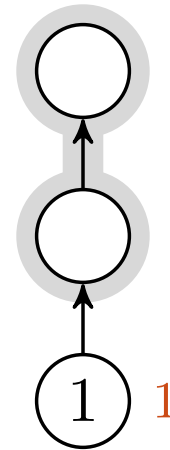
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

```
1 if  $n \geq 1$ 
```

```
2     return 1
```

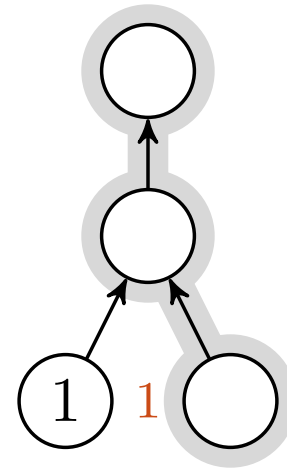
```
3  $m = \lfloor n/2 \rfloor$ 
```

```
4  $t_1 = \text{TIME}(m)$ 
```

```
5  $t_2 = \text{TIME}(n - m)$ 
```

```
6 return  $t_1 + t_2 + n$ 
```

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

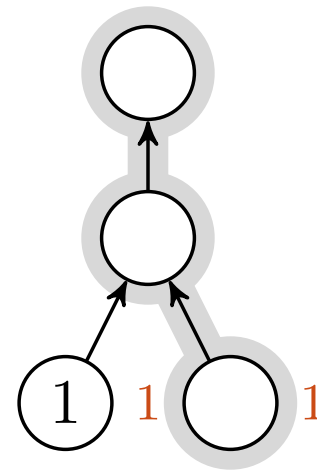
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

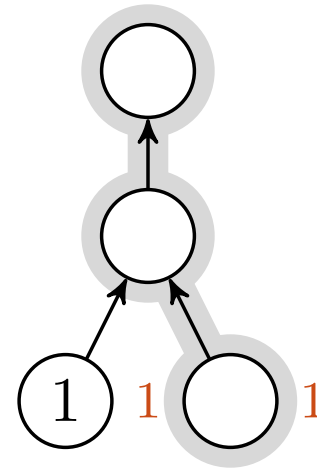
4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

→ 1

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

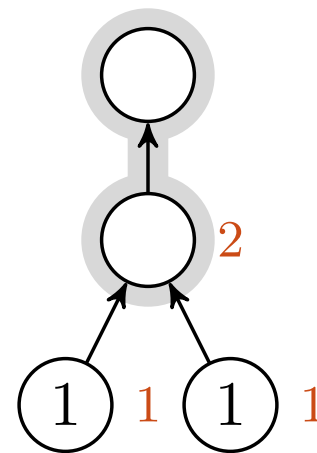
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

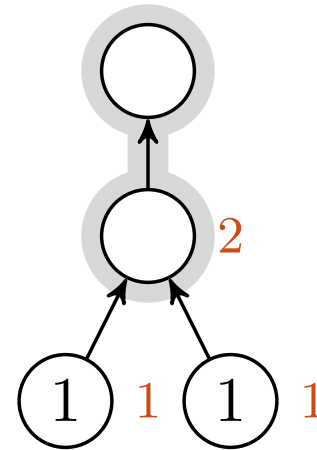
4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

→ 4

$n, m = 4, 2 \succ 2, 1$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

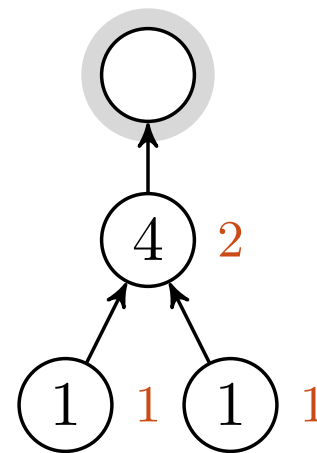
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

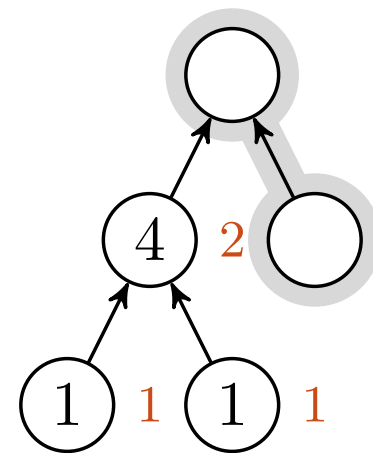
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

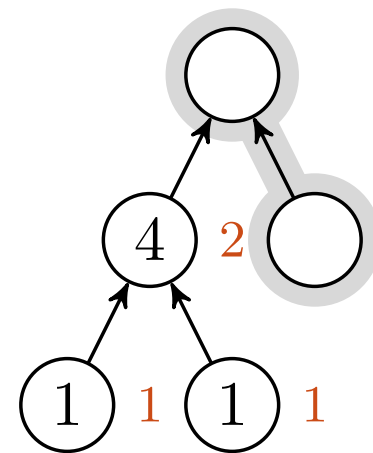
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

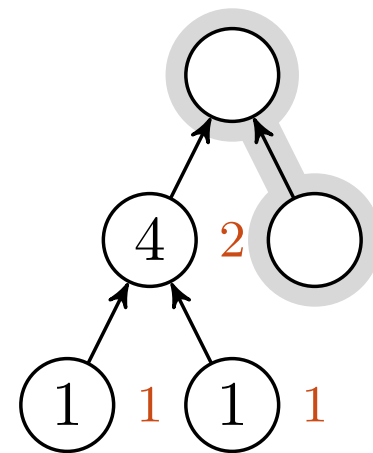
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

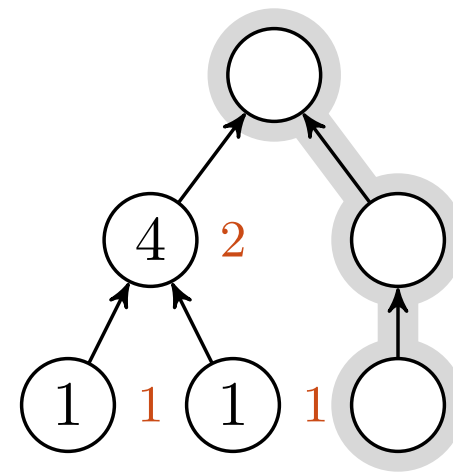
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

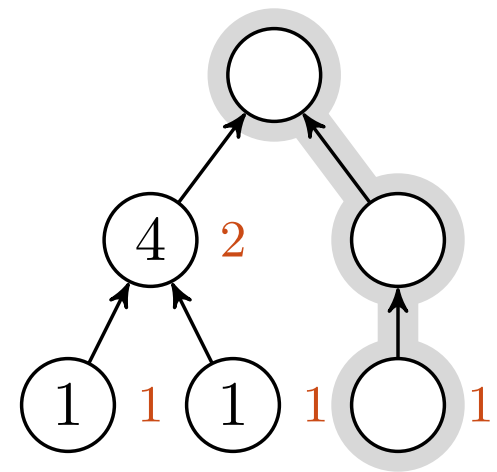
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

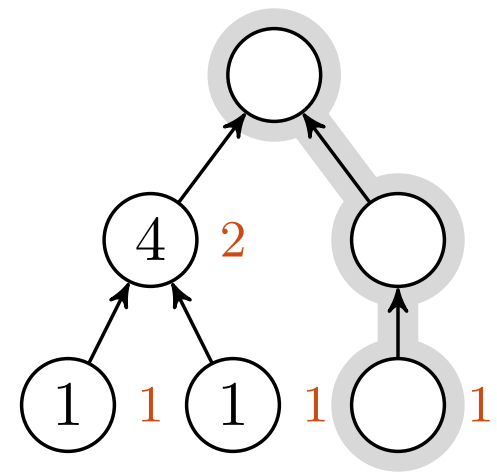
4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

→ 1

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

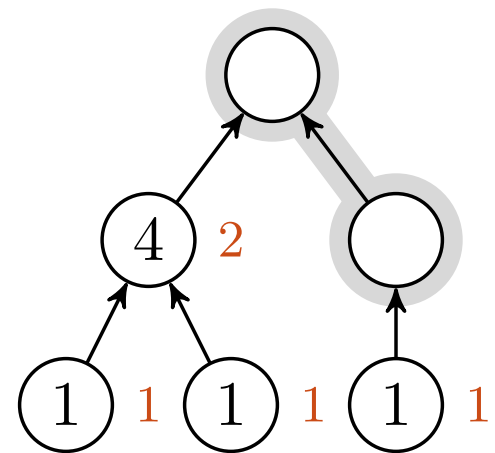
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

```
1 if  $n \geq 1$ 
```

```
2     return 1
```

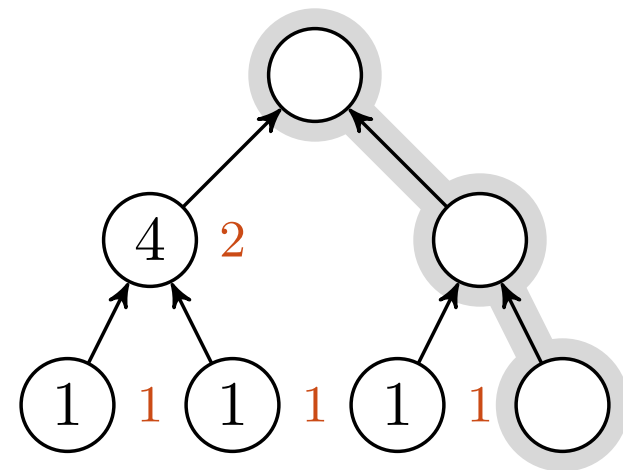
```
3  $m = \lfloor n/2 \rfloor$ 
```

```
4  $t_1 = \text{TIME}(m)$ 
```

```
5  $t_2 = \text{TIME}(n - m)$ 
```

```
6 return  $t_1 + t_2 + n$ 
```

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

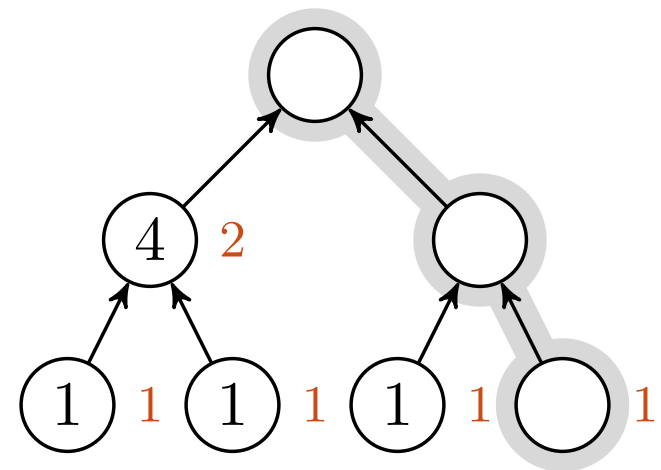
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

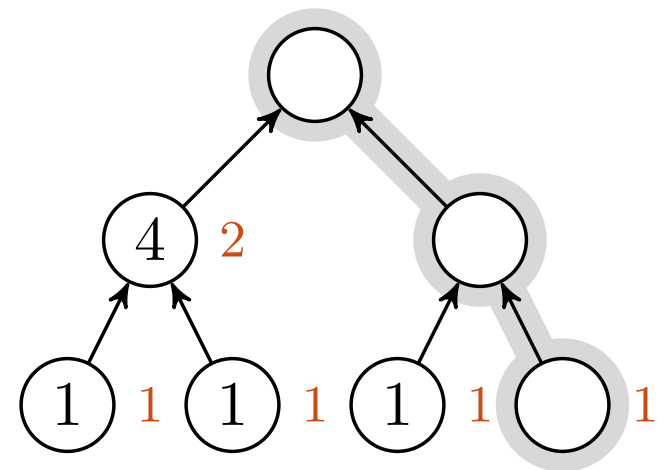
4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

→ 1

$n, m = 4, 2 \succ 2, 1 \succ 1, -$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

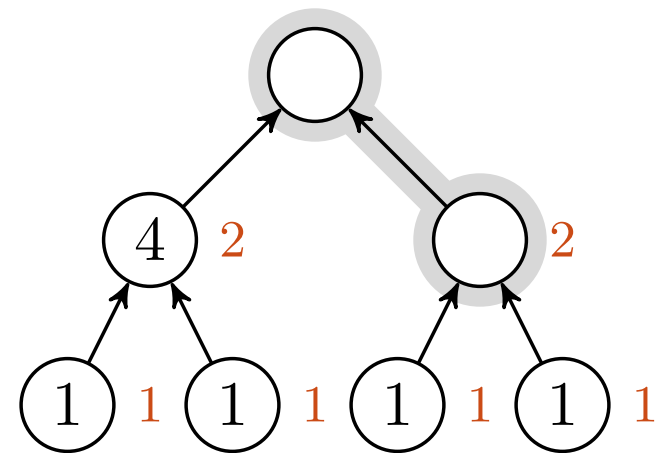
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2 \succ 2, 1$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

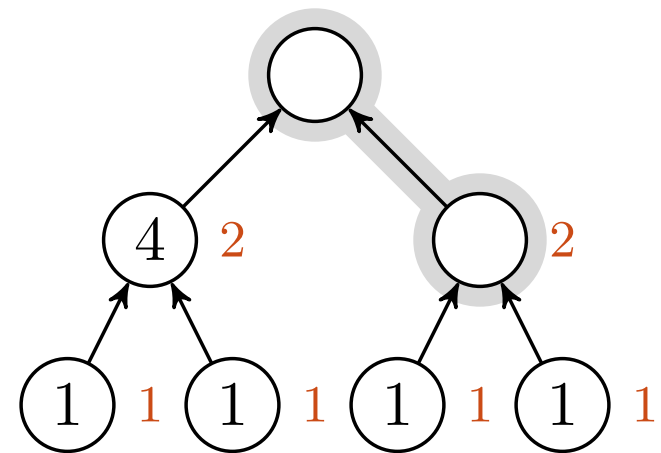
4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

→ 4

$n, m = 4, 2 \succ 2, 1$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

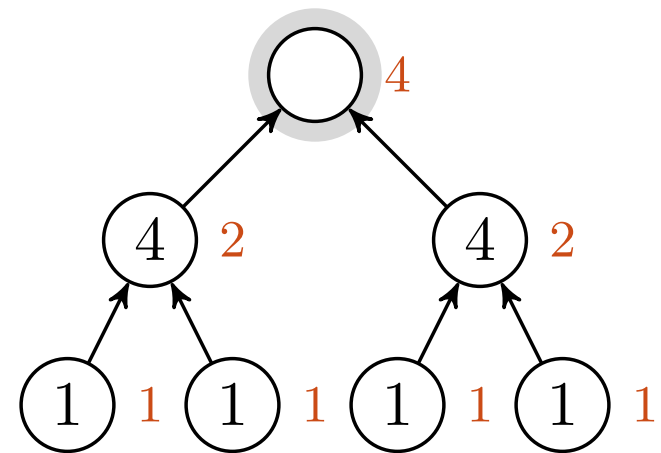
3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

$n, m = 4, 2$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

2 **return** 1

3 $m = \lfloor n/2 \rfloor$

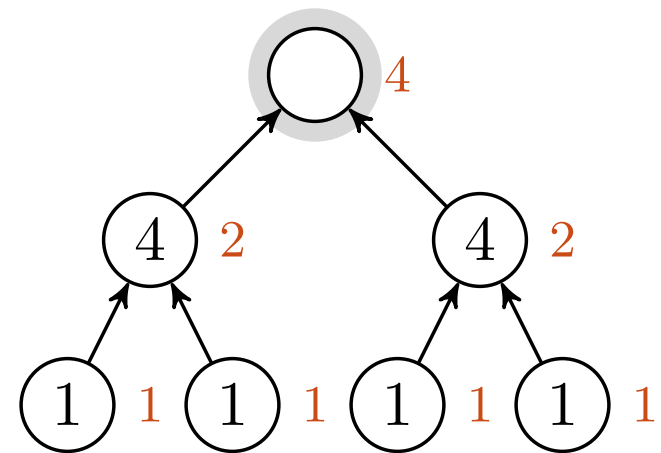
4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$

→ 12

$n, m = 4, 2$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

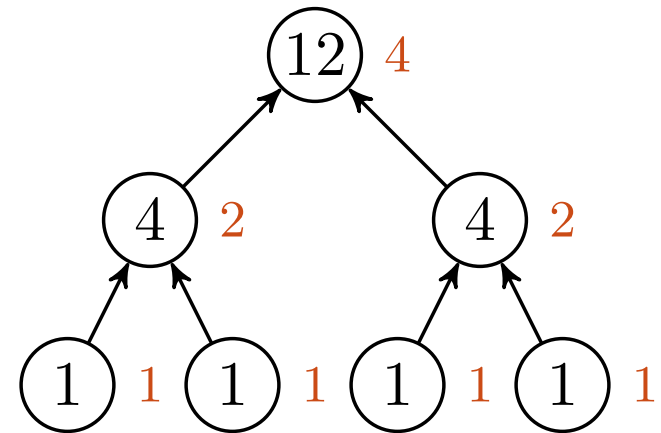
2 **return** 1

3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

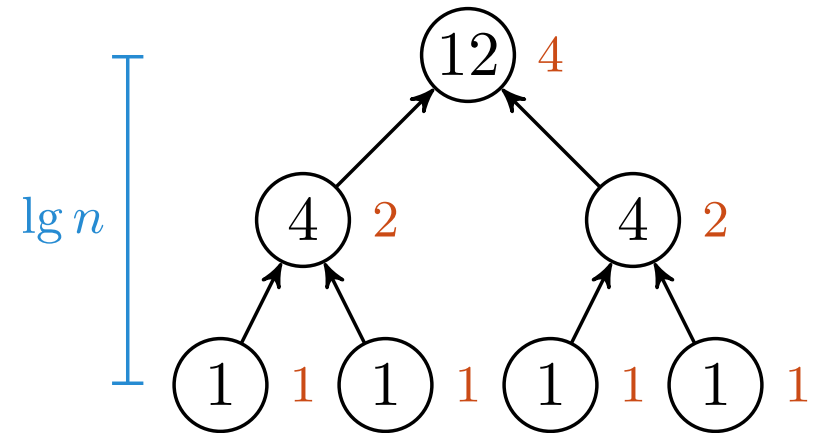
2 **return** 1

3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

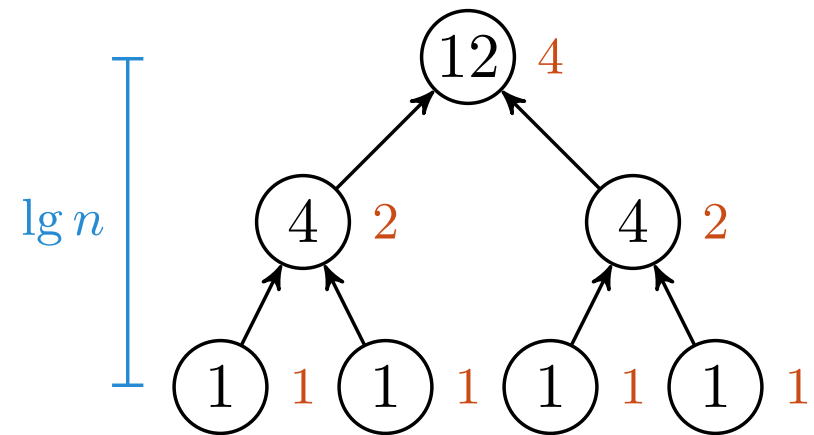
2 **return** 1

3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$



$$T(n) = n + \dots + n$$

$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

TIME(n)

1 **if** $n \geq 1$

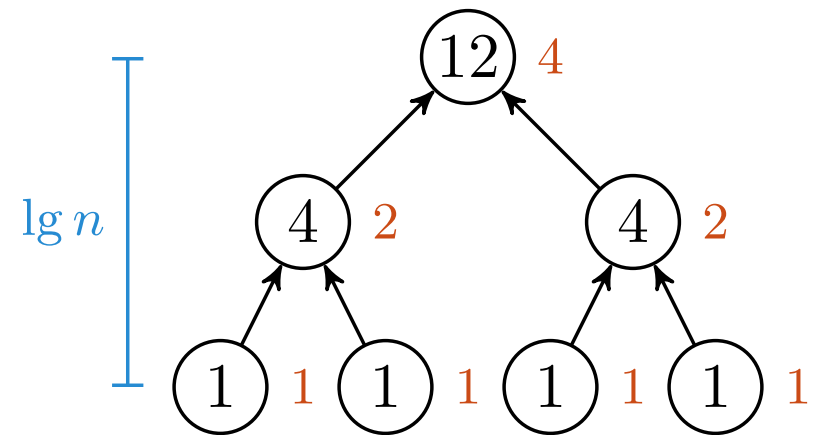
2 **return** 1

3 $m = \lfloor n/2 \rfloor$

4 $t_1 = \text{TIME}(m)$

5 $t_2 = \text{TIME}(n - m)$

6 **return** $t_1 + t_2 + n$



$$T(n) = n + \dots + n$$

$$T(n) = n \lg n + n$$

$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

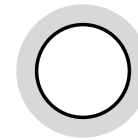
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

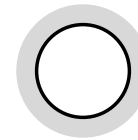
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

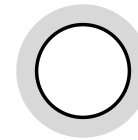
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

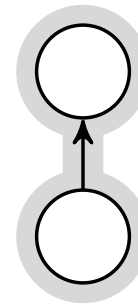
$$T(1) = 1$$

MERGE-SORT(A, p, r)

```
1  if  $p \geq r$ 
2      return
3   $q = \lfloor (p + r) / 2 \rfloor$ 
4  MERGE-SORT( $A, p, q$ )
5  MERGE-SORT( $A, q + 1, r$ )
6  MERGE( $A, p, q, r$ )
```

$$n = 4 \succ 2$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

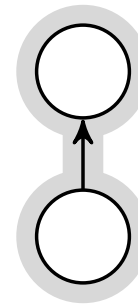
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 \succ 2$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

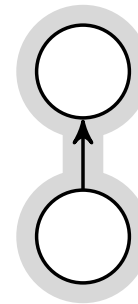
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 \succ 2$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

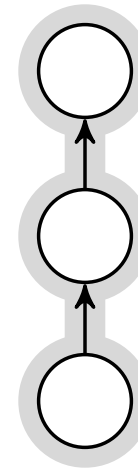
$$T(1) = 1$$

MERGE-SORT(A, p, r)

```
1  if  $p \geq r$ 
2      return
3   $q = \lfloor (p + r) / 2 \rfloor$ 
4  MERGE-SORT( $A, p, q$ )
5  MERGE-SORT( $A, q + 1, r$ )
6  MERGE( $A, p, q, r$ )
```

$$n = 4 \succ 2 \succ 1$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

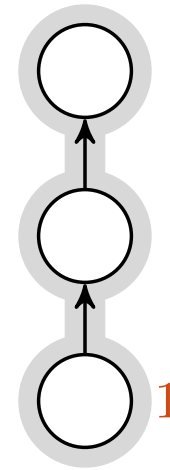
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 \succ 2 \succ 1$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

4 MERGE-SORT(A, p, q)

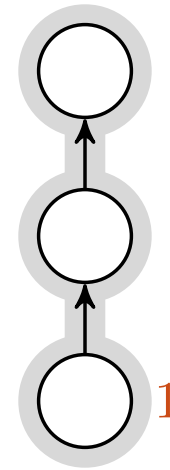
5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

→ 1

$n = 4 \succ 2 \succ 1$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

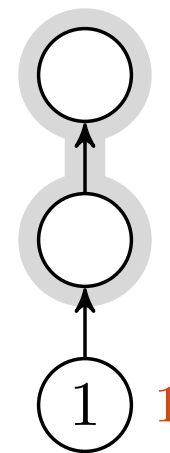
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 > 2$$

D&C > merge sort > rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

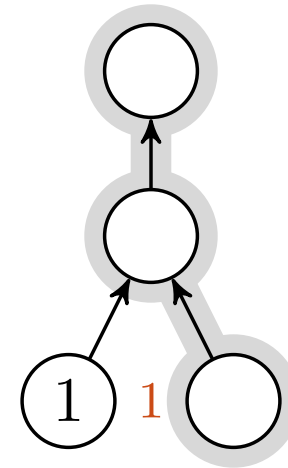
$$T(1) = 1$$

MERGE-SORT(A, p, r)

```
1  if  $p \geq r$ 
2      return
3   $q = \lfloor (p + r) / 2 \rfloor$ 
4  MERGE-SORT( $A, p, q$ )
5  MERGE-SORT( $A, q + 1, r$ )
6  MERGE( $A, p, q, r$ )
```

$n = 4 \succ 2 \succ 1$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

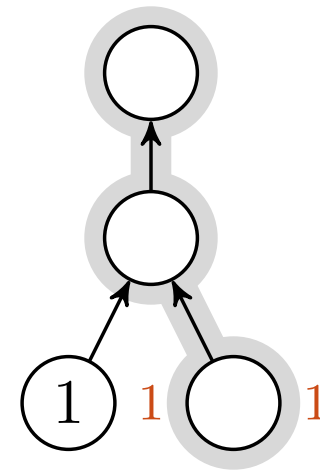
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 \succ 2 \succ 1$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

4 MERGE-SORT(A, p, q)

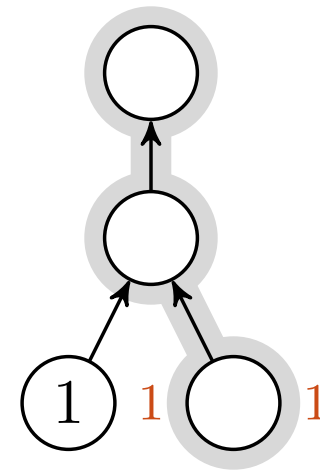
5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

→ 1

$n = 4 \succ 2 \succ 1$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

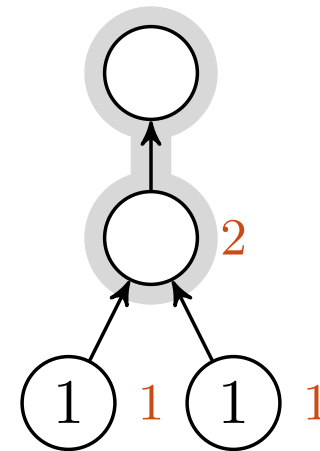
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 \succ 2$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r)/2 \rfloor$

4 MERGE-SORT(A, p, q)

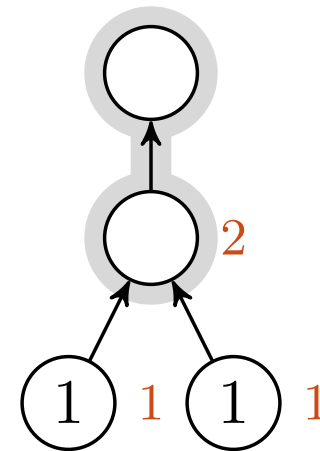
5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

→ 4

$$n = 4 \succ 2$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

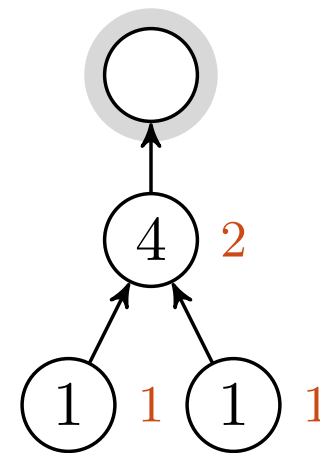
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

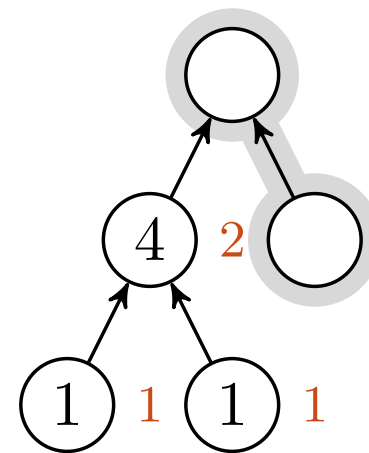
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 > 2$$

D&C > merge sort > rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

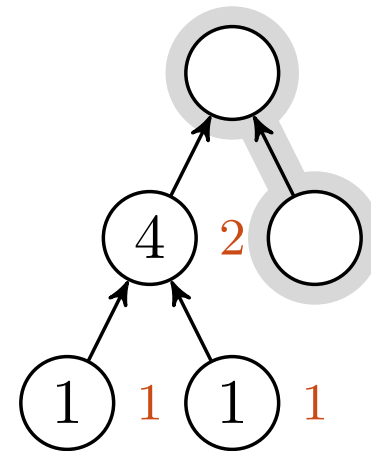
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 > 2$$

D&C > merge sort > rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

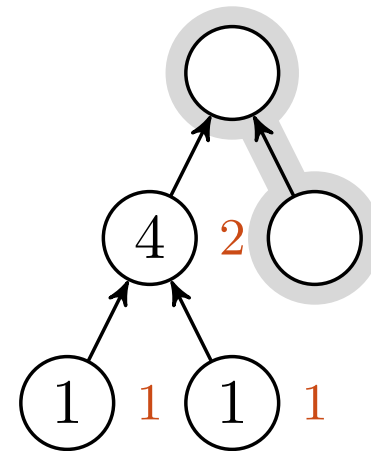
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 > 2$$

D&C > merge sort > rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

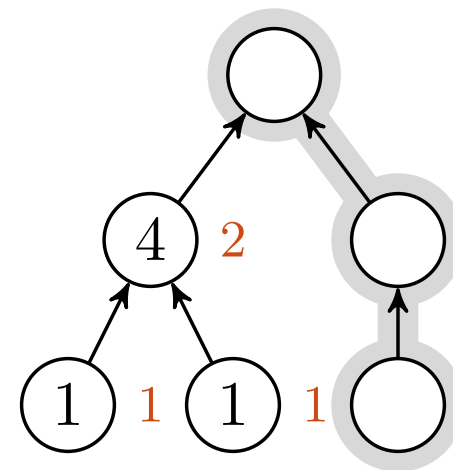
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 \succ 2 \succ 1$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r)/2 \rfloor$

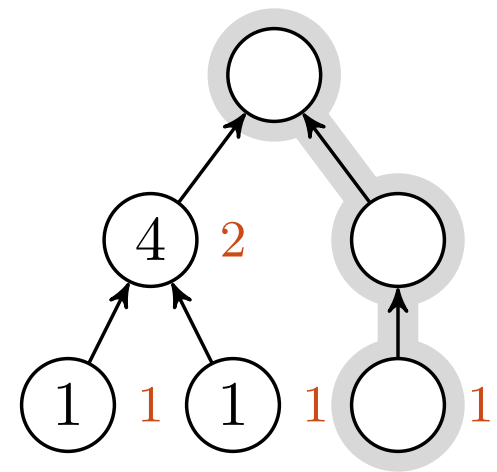
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 \succ 2 \succ 1$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

4 MERGE-SORT(A, p, q)

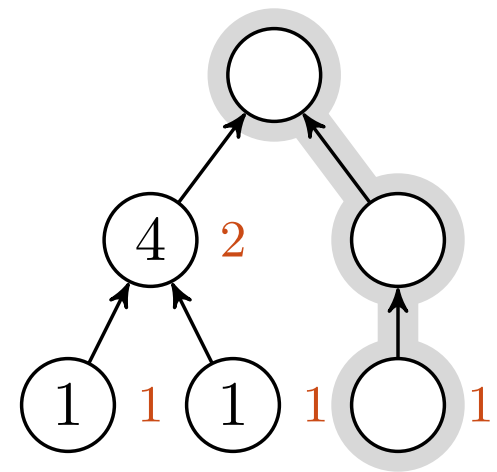
5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

→ 1

$n = 4 \succ 2 \succ 1$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r)/2 \rfloor$

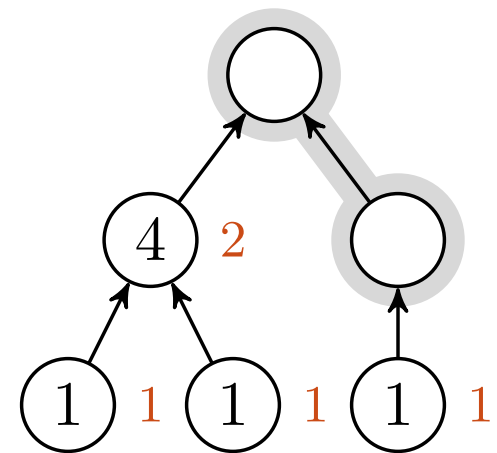
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 > 2$$

D&C > merge sort > rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

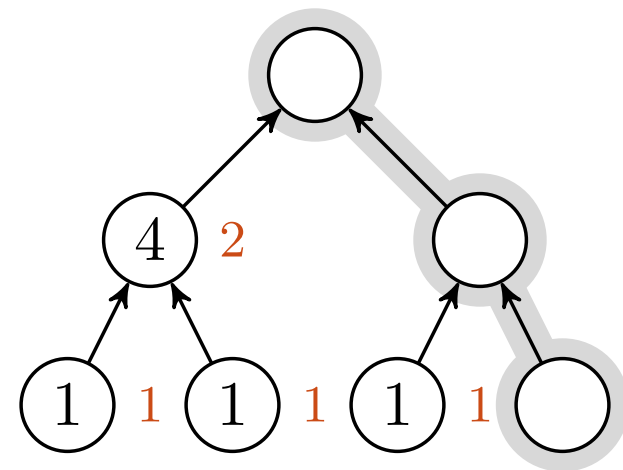
$$T(1) = 1$$

MERGE-SORT(A, p, r)

```
1 if  $p \geq r$   
2   return  
3    $q = \lfloor (p + r) / 2 \rfloor$   
4   MERGE-SORT( $A, p, q$ )  
5   MERGE-SORT( $A, q + 1, r$ )  
6   MERGE( $A, p, q, r$ )
```

$$n = 4 \succ 2 \succ 1$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r)/2 \rfloor$

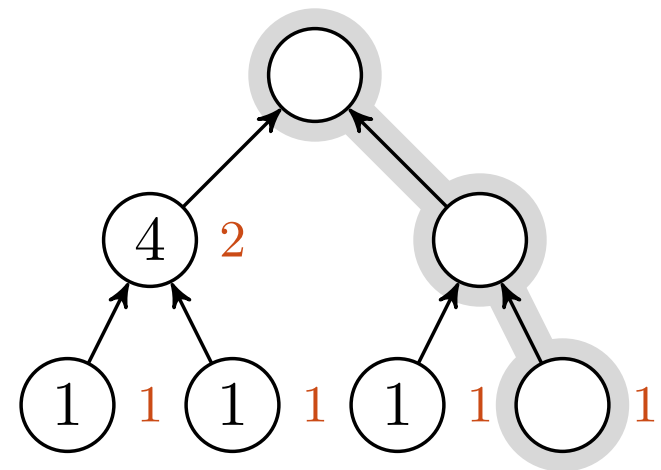
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 \succ 2 \succ 1$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r)/2 \rfloor$

4 MERGE-SORT(A, p, q)

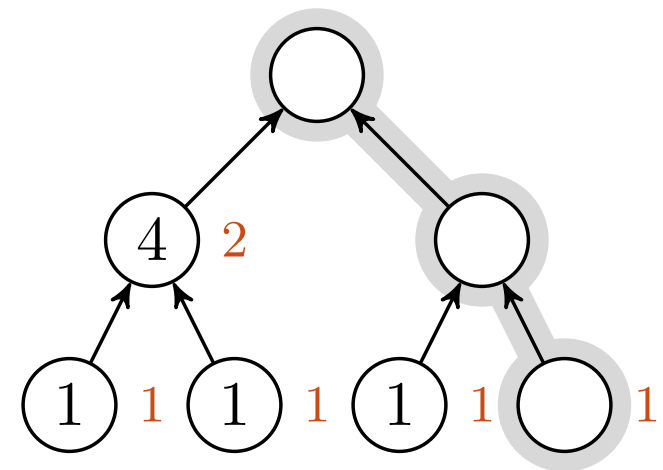
5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

→ 1

$$n = 4 \succ 2 \succ 1$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

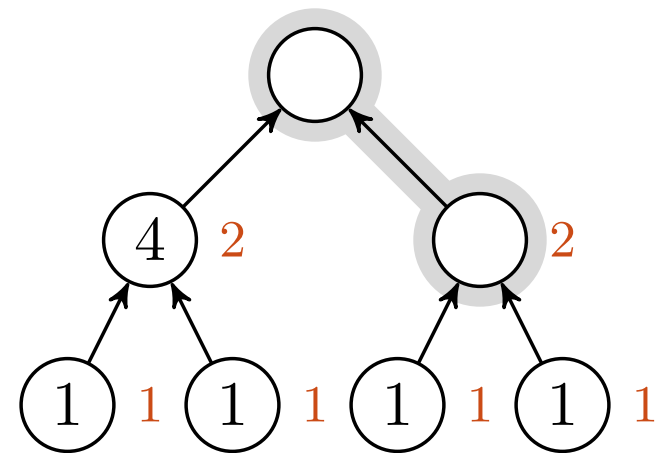
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4 > 2$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

4 MERGE-SORT(A, p, q)

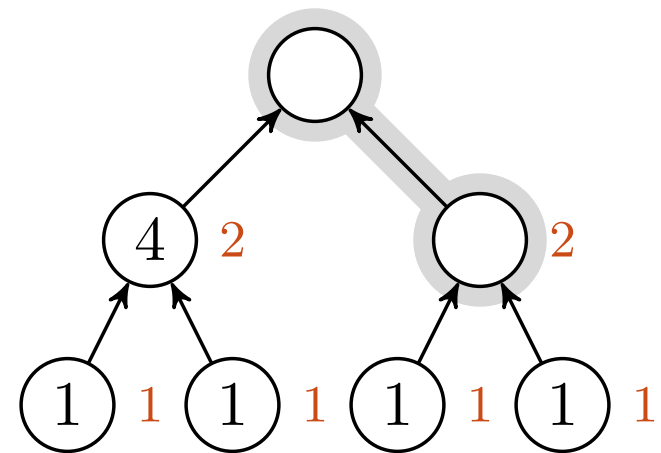
5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

→ 4

$$n = 4 > 2$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

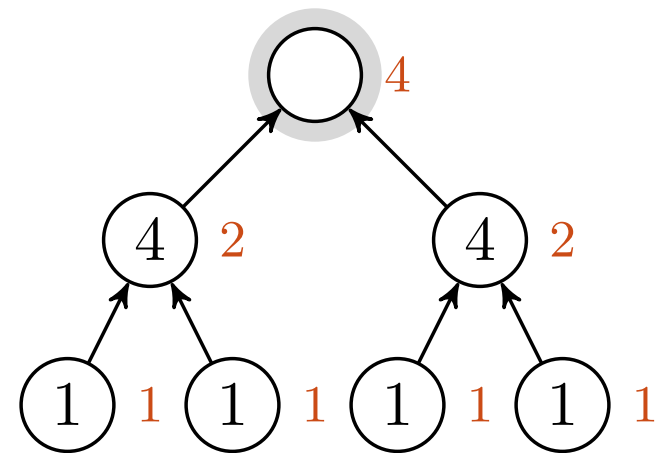
4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

$$n = 4$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

3 $q = \lfloor (p + r) / 2 \rfloor$

4 MERGE-SORT(A, p, q)

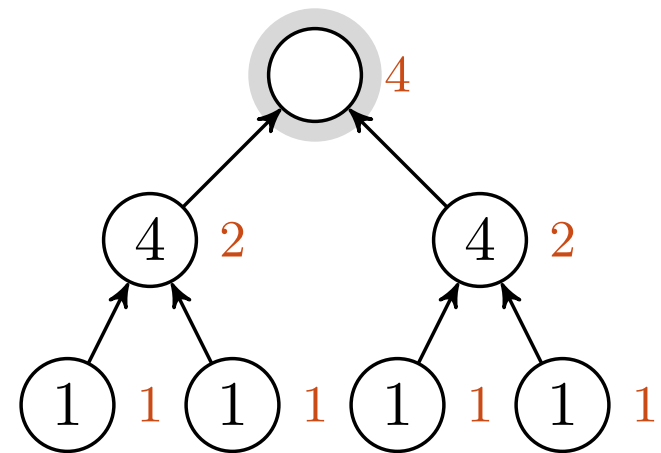
5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

→ 12

$$n = 4$$

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

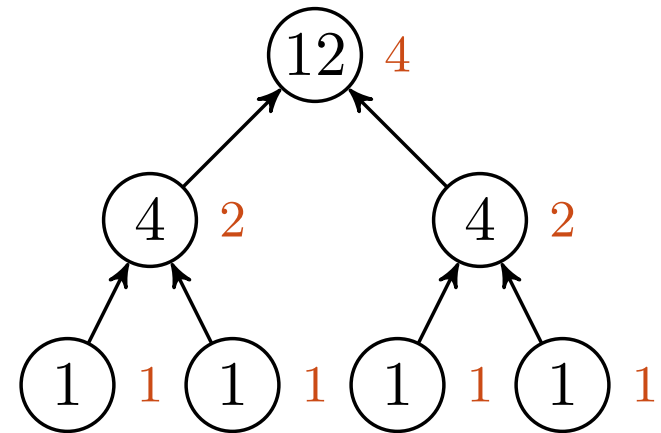
3 $q = \lfloor (p + r)/2 \rfloor$

4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

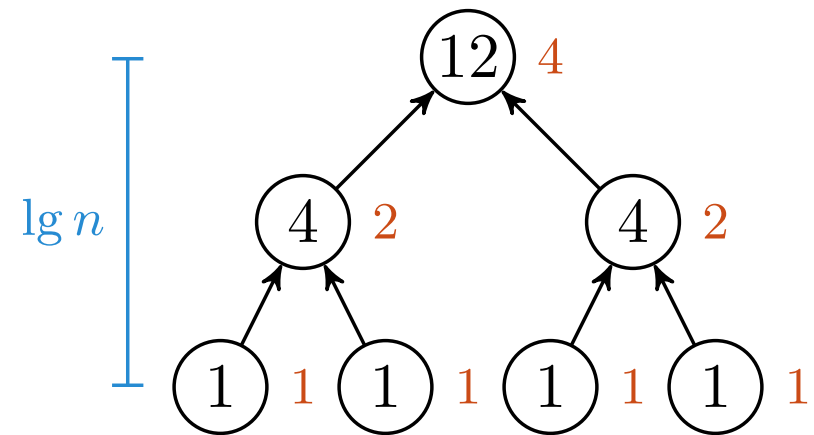
3 $q = \lfloor (p + r)/2 \rfloor$

4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

D&C › merge sort › rekurrenstre



$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

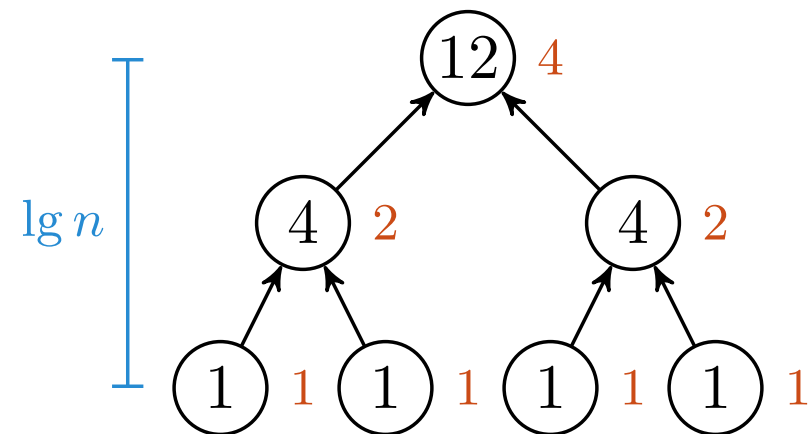
3 $q = \lfloor (p + r)/2 \rfloor$

4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

D&C › merge sort › rekurrenstre



$$T(n) = n + \dots + n$$

$$T(n) = 2 \cdot T(n/2) + n$$

$$T(1) = 1$$

MERGE-SORT(A, p, r)

1 **if** $p \geq r$

2 **return**

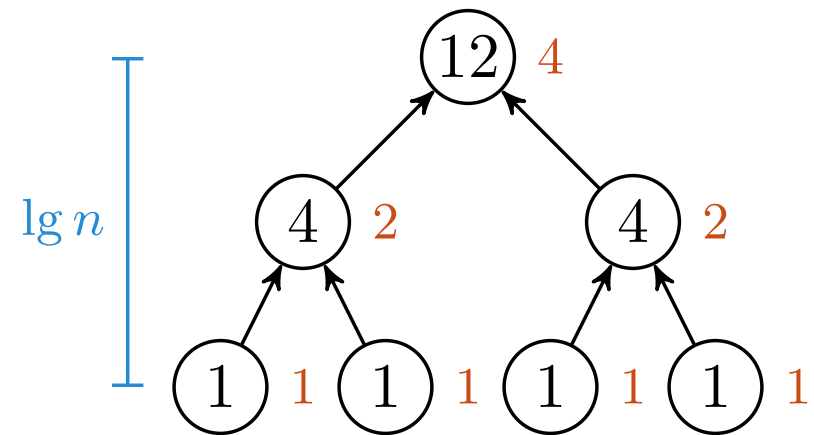
3 $q = \lfloor (p + r)/2 \rfloor$

4 MERGE-SORT(A, p, q)

5 MERGE-SORT($A, q + 1, r$)

6 MERGE(A, p, q, r)

D&C › merge sort › rekurrenstre



$$T(n) = n + \dots + n$$

$$T(n) = n \lg n + n$$