

Lecture 5

Merge Sort

Merge Pseudocode

— — —

Merge(A,p,q,r):

$n_1 = q - p + 1$

$n_2 = r - q$

Let $L[1 \dots n_1 + 1]$ and $R[1 \dots n_2 + 1]$ be new arrays

for $i = 1$ to n_1 :

$L[i] = A[p + i - 1]$

for $i = 1$ to n_2 :

$R[i] = A[q + i]$

$L[n_1 + 1] = \infty$

$R[n_2 + 1] = \infty$

Merge Pseudocode Continued.

— — —

```
i=1
```

```
j=1
```

```
for k=p to r:
```

```
    if  $L[i] \leq R[j]$ :
```

```
         $A[k] = L[i]$ 
```

```
         $i = i + 1$ 
```

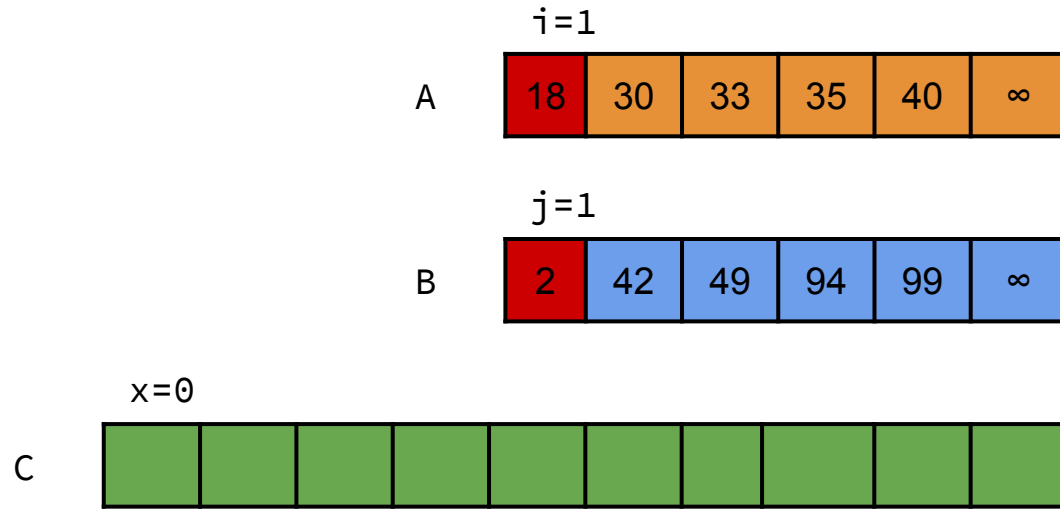
```
    else:
```

```
         $A[k] = R[j]$ 
```

```
         $j = j + 1$ 
```

Simulation of Merge

— — —



A

18	30	33	35	40	∞
----	----	----	----	----	----------

B

2	42	49	94	99	∞
---	----	----	----	----	----------

C



Simulation of Merge

— — —

Comparing $A[i]$ with $B[j]$ and
setting $C[x] = \min(A[i], B[j])$

$i=1$

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

$j=2$

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

$x=1$

C	2								
---	---	--	--	--	--	--	--	--	--

Simulation of Merge

— — —

Comparing $A[i]$ with $B[j]$ and
setting $C[x] = \min(A[i], B[j])$

$i=2$

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

$j=2$

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

$x=2$

C	2	18							
---	---	----	--	--	--	--	--	--	--

Simulation of Merge

— — —

Comparing $A[i]$ with $B[j]$ and
setting $C[x] = \min(A[i], B[j])$

$i=3$

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

$j=2$

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

$x=3$

C	2	18	30						
---	---	----	----	--	--	--	--	--	--

Simulation of Merge

— — —

Comparing $A[i]$ with $B[j]$ and
setting $C[x] = \min(A[i], B[j])$

$i=4$

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

$j=2$

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

$x=4$

C	2	18	30	33						
---	---	----	----	----	--	--	--	--	--	--

Simulation of Merge

Comparing $A[i]$ with $B[j]$ and
setting $C[x] = \min(A[i], B[j])$

$i=5$

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

$j=2$

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

$x=5$

C	2	18	30	33	35					
---	---	----	----	----	----	--	--	--	--	--

Simulation of Merge

Comparing $A[i]$ with $B[j]$ and
setting $C[x] = \min(A[i], B[j])$

$i=6$

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

$j=2$

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

$x=6$

C	2	18	30	33	35	40				
---	---	----	----	----	----	----	--	--	--	--

Simulation of Merge

As one of the auxiliary array reached the sentinel value ∞ therefore, copying the rest of the values to the array C

i=6

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

j=3

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

x=7

C	2	18	30	33	35	40	42			
---	---	----	----	----	----	----	----	--	--	--

Simulation of Merge

As one of the auxiliary array reached the sentinel value ∞ therefore, copying the rest of the values to the array C

i=6

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

j=4

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

x=8

C	2	18	30	33	35	40	42	49		
---	---	----	----	----	----	----	----	----	--	--

Simulation of Merge

As one of the auxiliary array reached the sentinel value ∞ therefore, copying the rest of the values to the array C

i=6

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

j=5

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

x=9

C	2	18	30	33	35	40	42	49	94	
---	---	----	----	----	----	----	----	----	----	--

Simulation of Merge

As one of the auxiliary array reached the sentinel value ∞ therefore, copying the rest of the values to the array C

i=6

A	18	30	33	35	40	∞
---	----	----	----	----	----	----------

j=6

B	2	42	49	94	99	∞
---	---	----	----	----	----	----------

x=10

C	2	18	30	33	35	40	42	49	94	99
---	---	----	----	----	----	----	----	----	----	----

Merge Sort Pseudocode

— — —

Merge_Sort(A,p,r):

 if $p < r$:

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

Merge_Sort(A,p,q)

Merge_Sort(A,q+1,r)

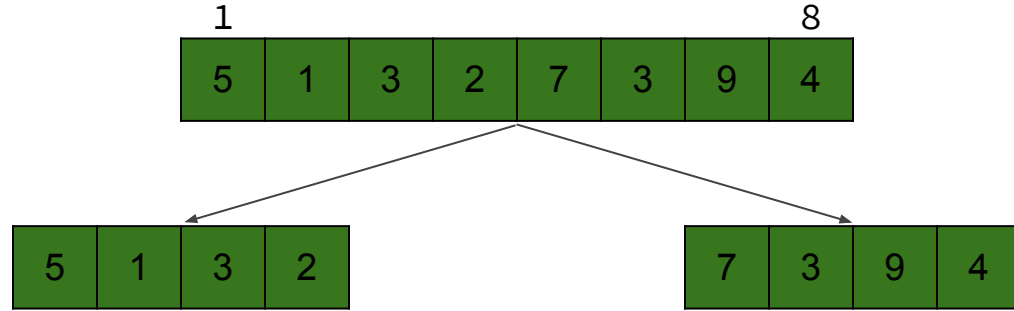
Merge(A,p,q,r)

Simulation of Merge Sort

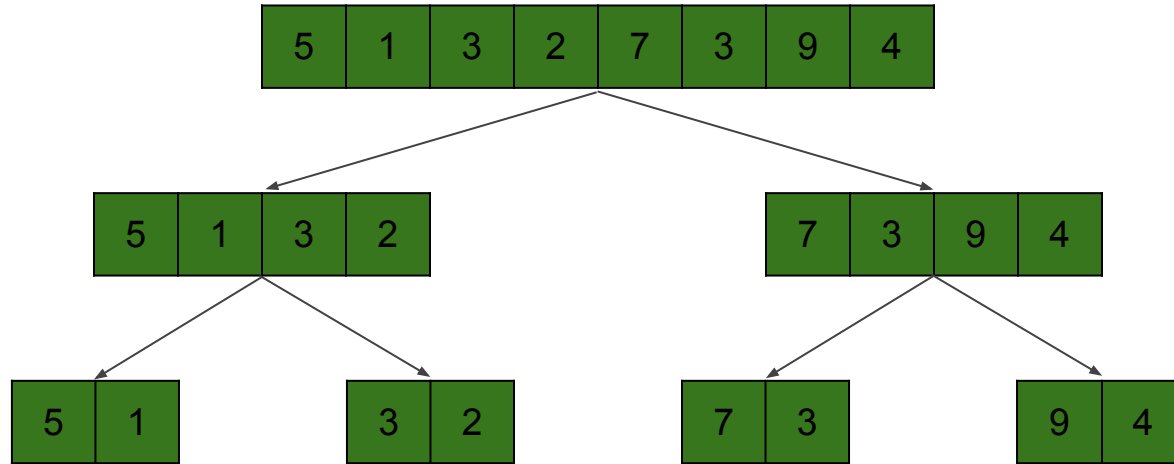
--

1				8			
5	1	3	2	7	3	9	4

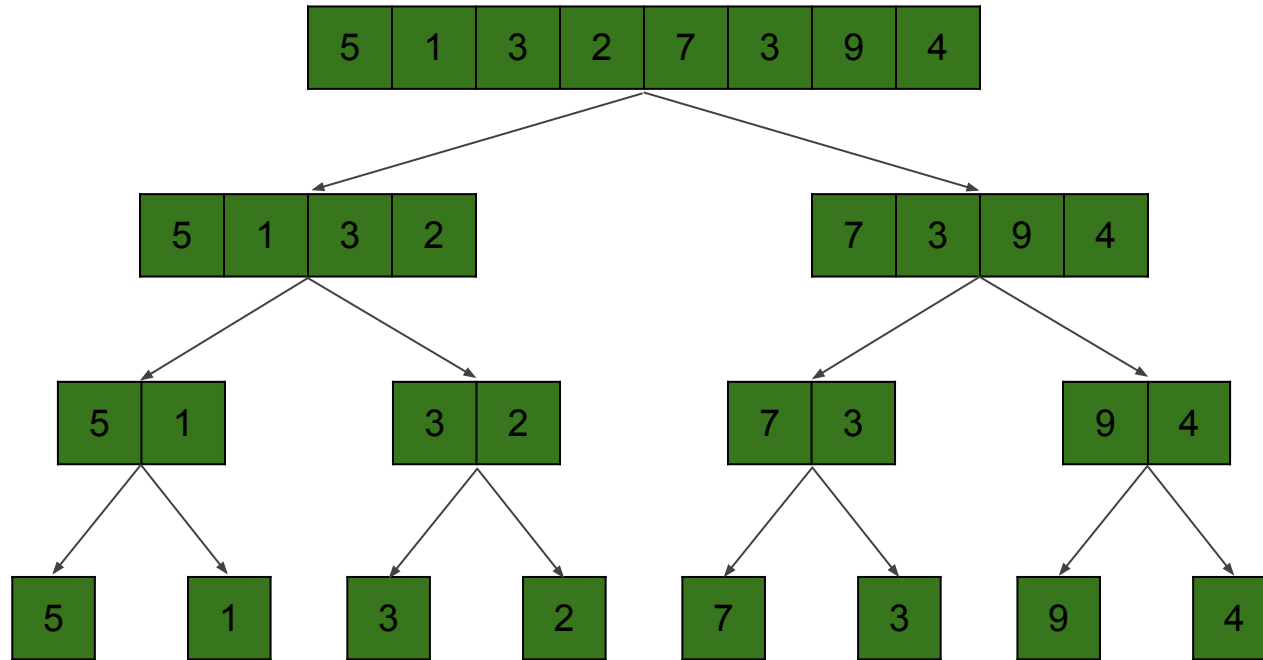
Simulation of Merge Sort



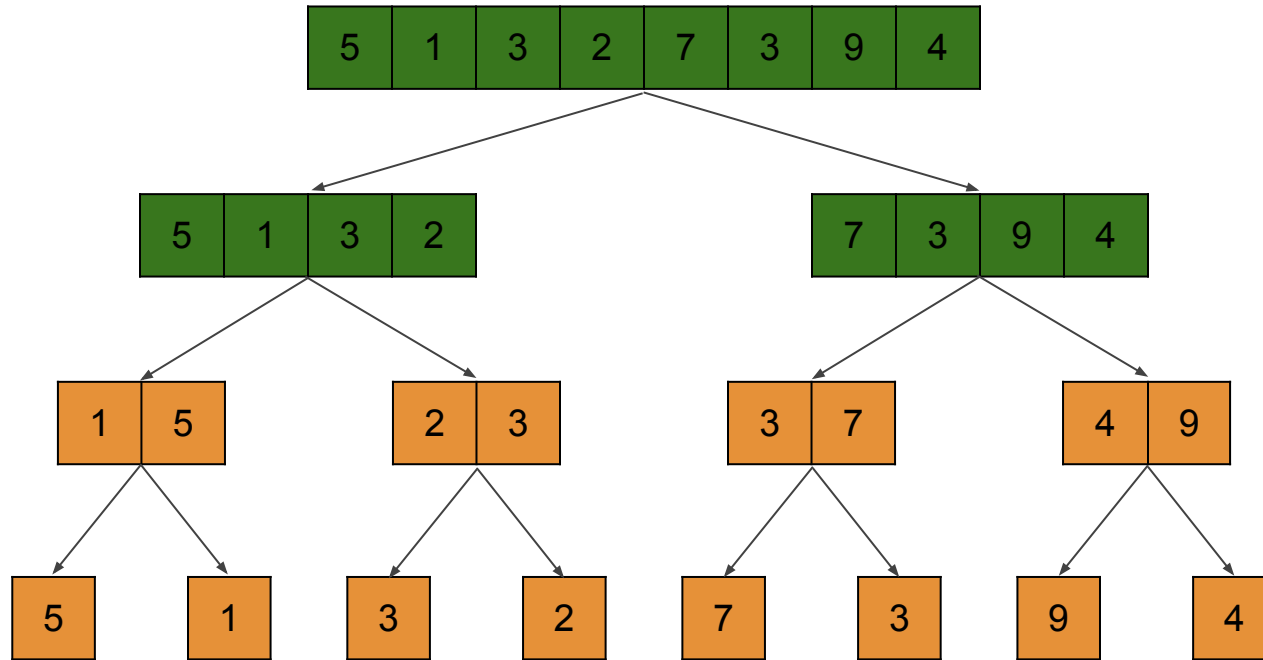
Simulation of Merge Sort



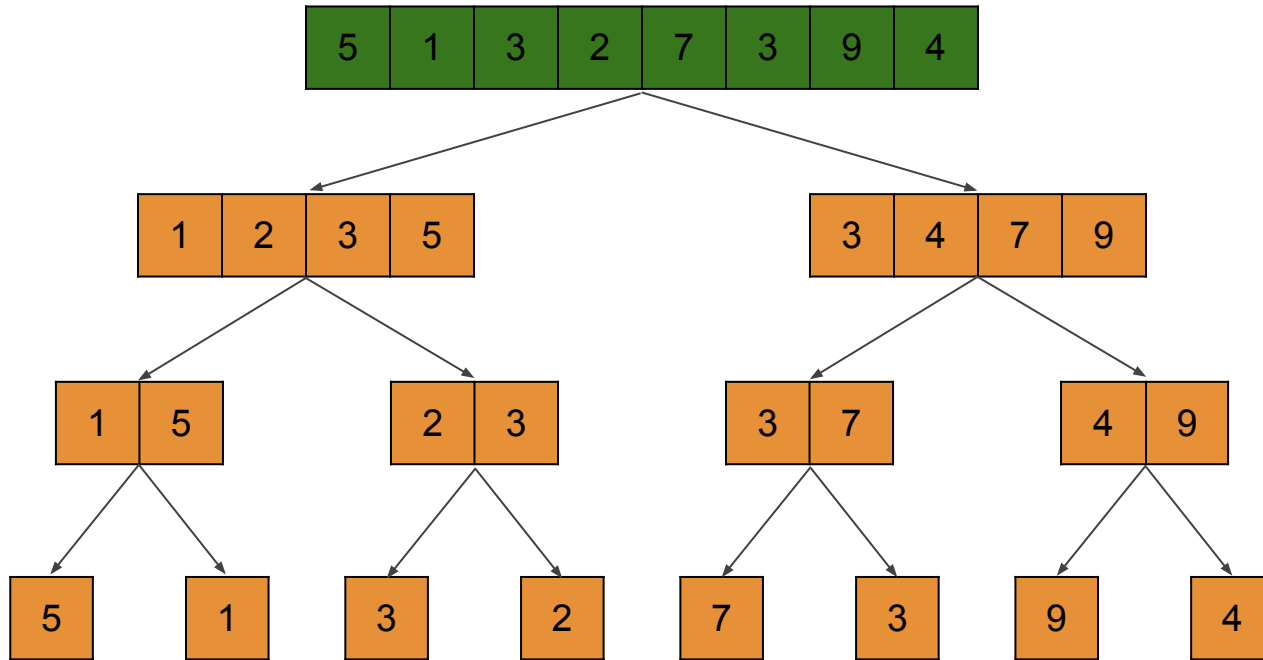
Simulation of Merge Sort



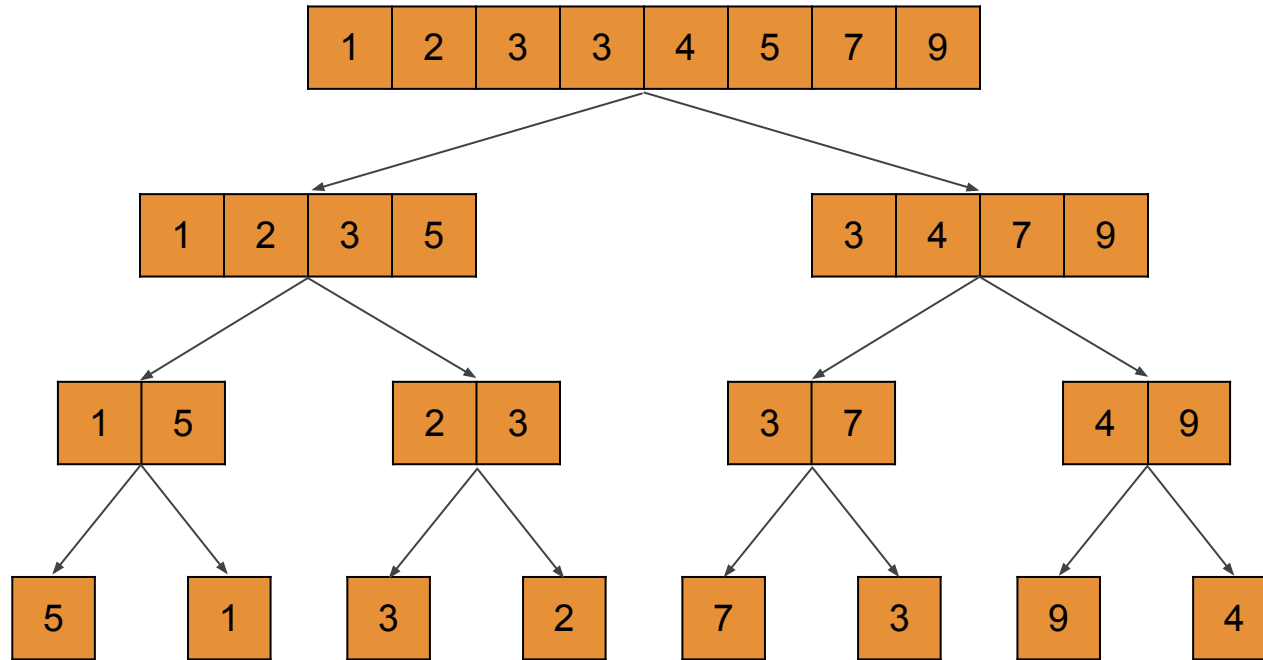
Simulation of Merge Sort



Simulation of Merge Sort



Simulation of Merge Sort



Analysis of Merge Sort

The height h of the merge-sort tree is $O(\log_2 n)$. At each recursive call we divide in half the sequence, The overall amount of work done at the nodes of depth i is $O(n)$. We partition and merge 2^i sequences of size $n/2^i$ and we make 2^{i+1} recursive call Thus, the total running time of merge-sort is $O(n \log_2 n)$

Analysis of Merge Sort Continued

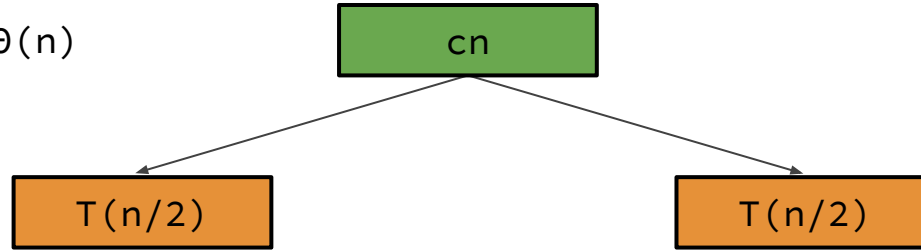
— — —

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

$T(n)$

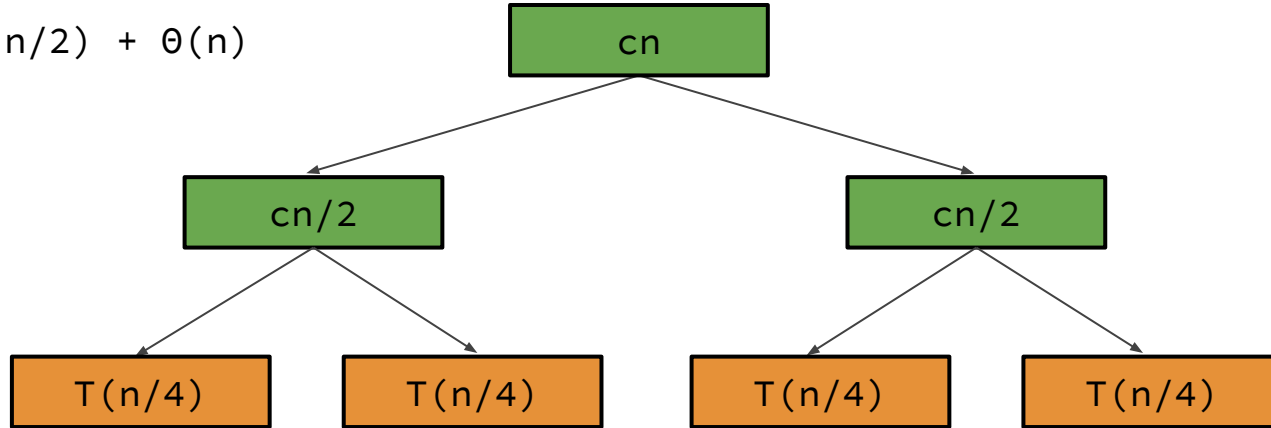
Analysis of Merge Sort Continued

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



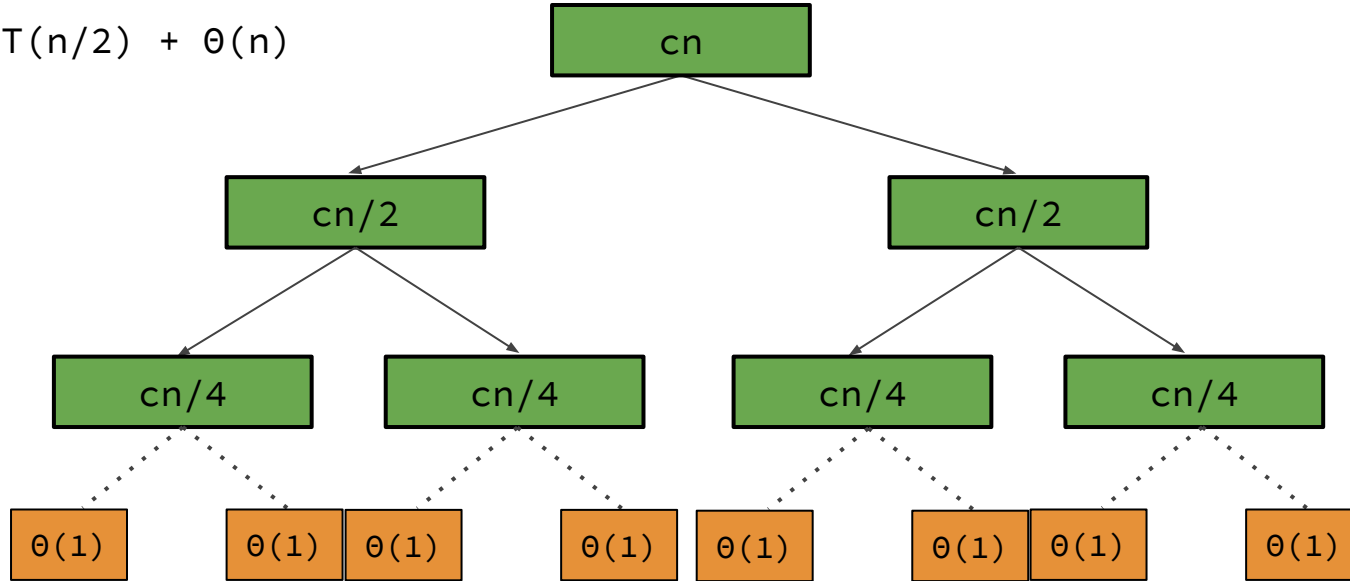
Analysis of Merge Sort Continued

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



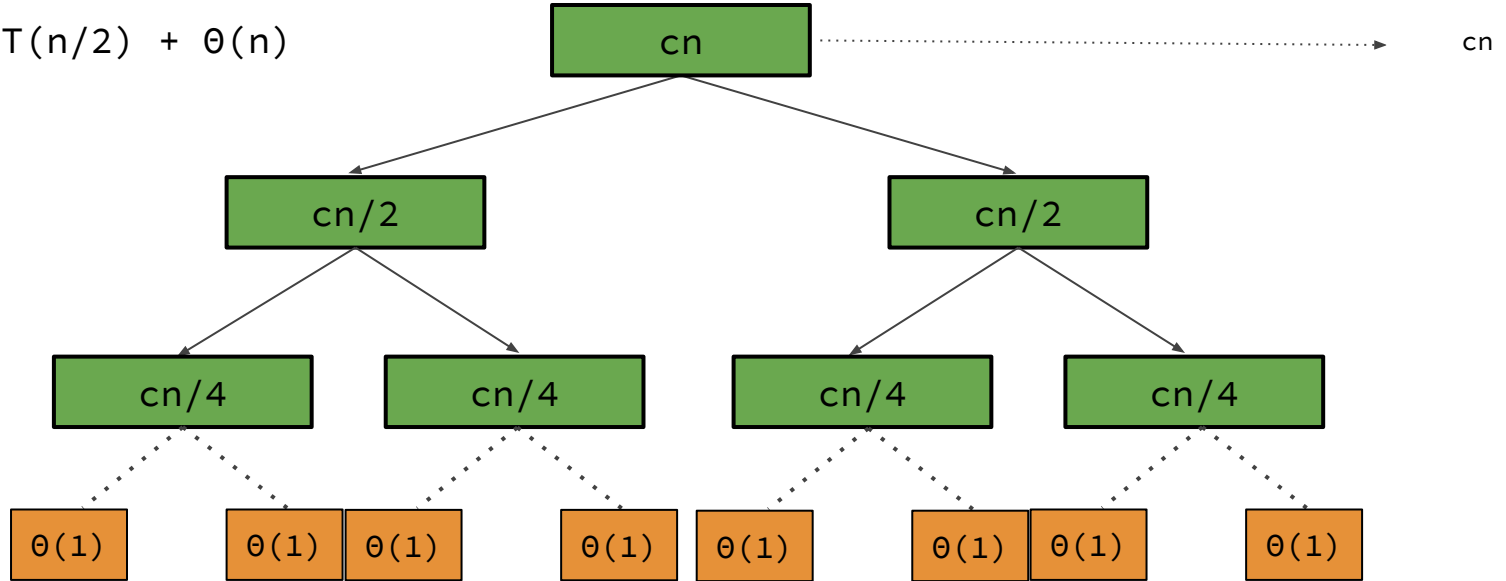
Analysis of Merge Sort Continued

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



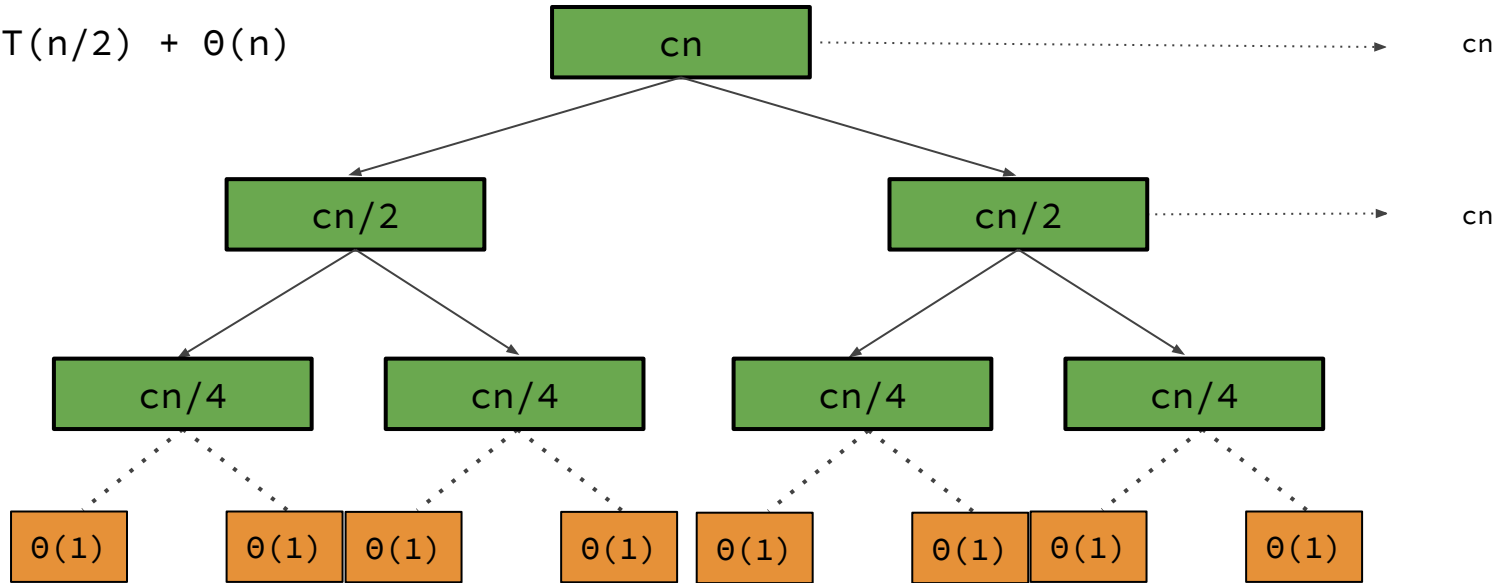
Analysis of Merge Sort Continued

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



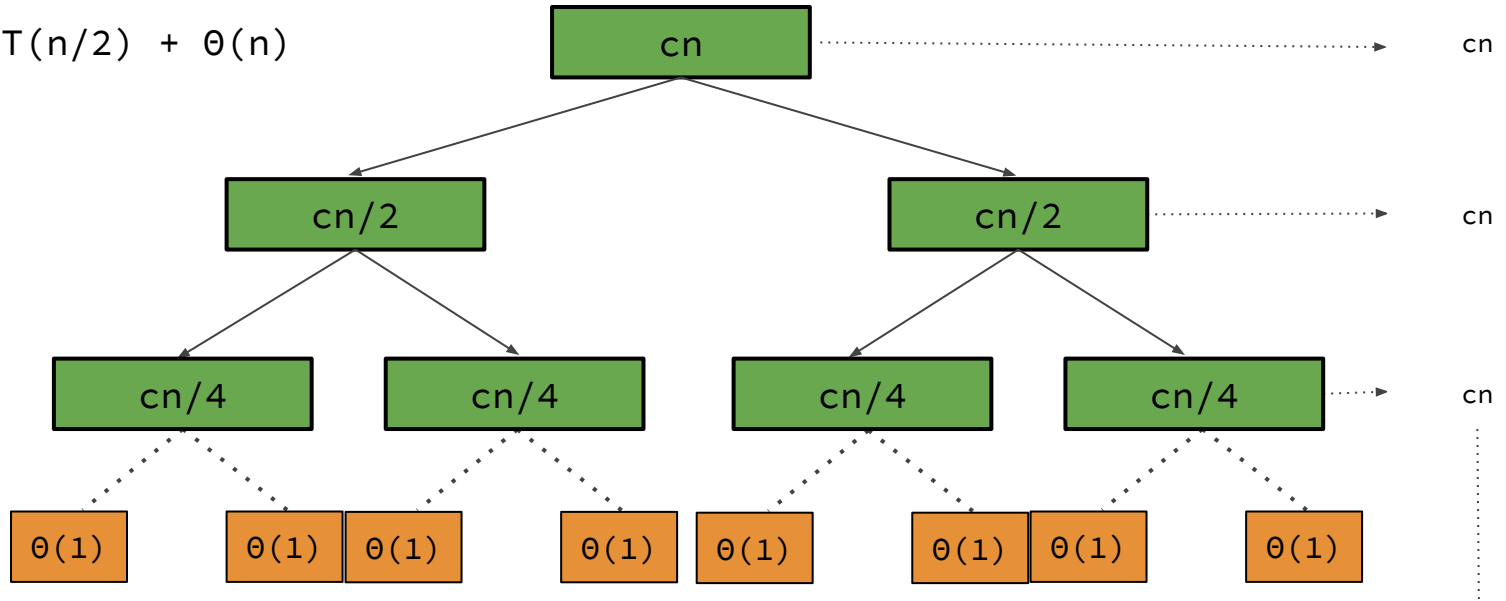
Analysis of Merge Sort Continued

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



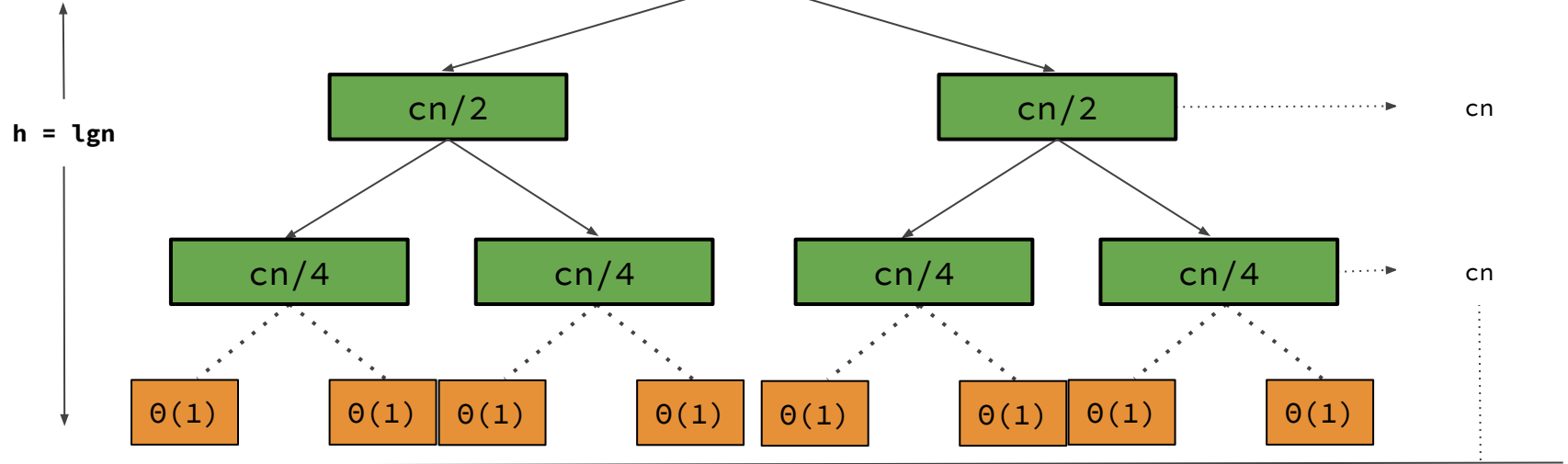
Analysis of Merge Sort Continued

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



Analysis of Merge Sort Continued

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



$$\begin{aligned} T(n) &= \lg n * cn \\ &= \theta(n \lg n) \end{aligned}$$