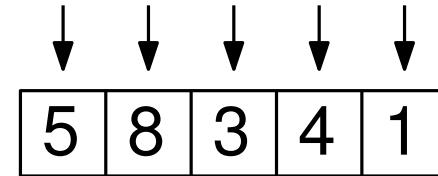
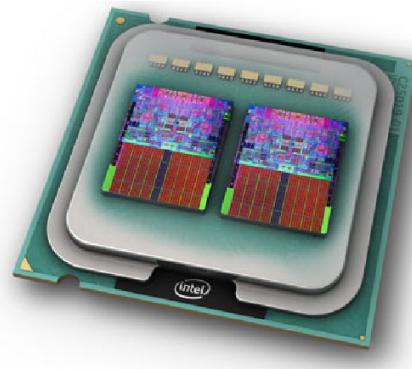




ICS 643: Advanced Parallel Algorithms

Prof. Nodari Sitchinava



Lecture 4: Prefix Sums

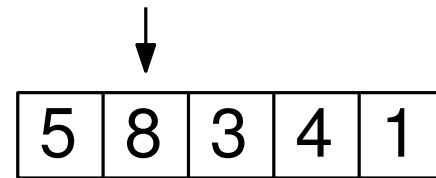
Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```

5	8	3	4	1
---	---	---	---	---

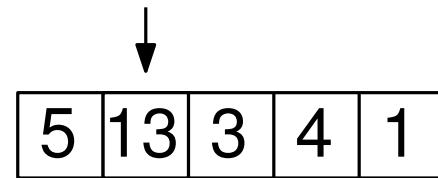
Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```



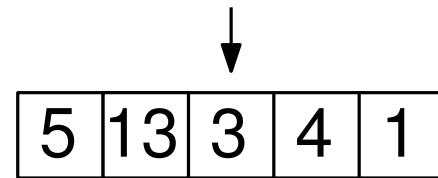
Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```



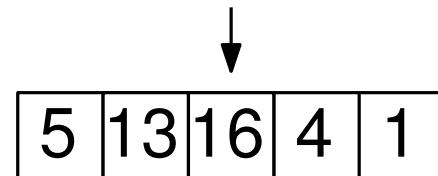
Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```



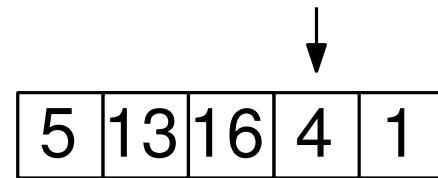
Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```



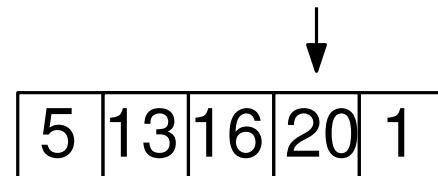
Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```



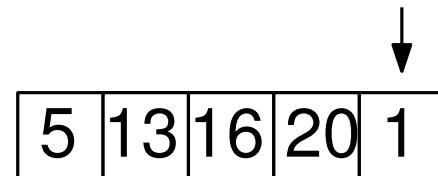
Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```



Prefix Sums

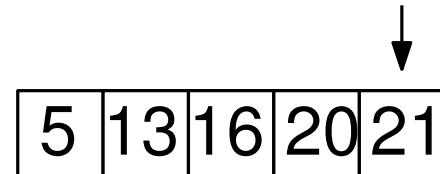
```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```



5	13	16	20	1
---	----	----	----	---

Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```



A horizontal row of five boxes containing the numbers 5, 13, 16, 20, and 21. An arrow points downwards from the top right towards the number 21.

5	13	16	20	21
---	----	----	----	----

Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```

5	13	16	20	21
---	----	----	----	----

Time

$O(n)$

Prefix Sums

5	8	3	4	1
---	---	---	---	---

Time

5	13	16	20	21
---	----	----	----	----

$O(n)$

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```

Prefix Sums

5	8	3	4	1
---	---	---	---	---

Time

5	13	16	20	21
---	----	----	----	----

$O(n)$

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```

```
for  $i = 2$  to  $n$  in parallel do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```

5	8	3	4	1
---	---	---	---	---

Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$ 
```

```
return  $a[n]$ 
```

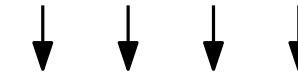
```
for  $i = 2$  to  $n$  in parallel do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```

5	8	3	4	1
---	---	---	---	---

Time

5	13	16	20	21
---	----	----	----	----

$O(n)$



5	8	3	4	1
---	---	---	---	---

Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$ 
```

```
return  $a[n]$ 
```

```
for  $i = 2$  to  $n$  in parallel do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```

5	8	3	4	1
---	---	---	---	---

Time

5	13	16	20	21
---	----	----	----	----

$O(n)$

5	8	3	4	1
---	---	---	---	---

Start $n - 1$ threads t_2, \dots, t_n
Each thread t_i (where $i = 2, \dots, n$) **do**:
 READ $r_{0,i} \leftarrow a[i]$
 EXECUTE
 WRITE
 READ $r_{1,i} \leftarrow a[i - 1]$
 EXECUTE $r_{0,i} \leftarrow r_{0,i} + r_{1,i}$
 WRITE $a[i] \leftarrow r_{0,i}$

Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$ 
```

```
return  $a[n]$ 
```

```
for  $i = 2$  to  $n$  in parallel do
```

```
     $a[i] = a[i] + a[i - 1]$ 
```

```
return  $a[n]$ 
```

5	8	3	4	1
---	---	---	---	---

Time

5	13	16	20	21
---	----	----	----	----

$O(n)$

5	13	11	7	5
---	----	----	---	---

Start $n - 1$ threads t_2, \dots, t_n
Each thread t_i (where $i = 2, \dots, n$) **do**:

```
    READ  $r_{0,i} \leftarrow a[i]$   
    EXECUTE  
    WRITE  
    READ  $r_{1,i} \leftarrow a[i - 1]$   
    EXECUTE  $r_{0,i} \leftarrow r_{0,i} + r_{1,i}$   
    WRITE  $a[i] \leftarrow r_{0,i}$ 
```

Prefix Sums

```
for  $i = 2$  to  $n$  do  
     $a[i] = a[i] + a[i - 1]$ 
```

```
return  $a[n]$ 
```

```
for  $i = 2$  to  $n$  in parallel do  
     $a[i] = a[i] + a[i - 1]$   
return  $a[n]$ 
```

5	8	3	4	1
---	---	---	---	---

Time

5	13	16	20	21
---	----	----	----	----

$O(n)$

5	13	11	7	5
---	----	----	---	---

$O(1)$

Start $n - 1$ threads t_2, \dots, t_n
Each thread t_i (where $i = 2, \dots, n$) **do**:

```
    READ  $r_{0,i} \leftarrow a[i]$   
    EXECUTE  
    WRITE  
    READ  $r_{1,i} \leftarrow a[i - 1]$   
    EXECUTE  $r_{0,i} \leftarrow r_{0,i} + r_{1,i}$   
    WRITE  $a[i] \leftarrow r_{0,i}$ 
```

Parallel Prefix Sums

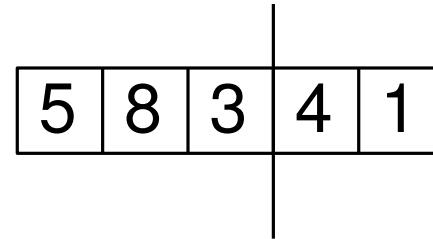
5	8	3	4	1
---	---	---	---	---

Parallel Prefix Sums

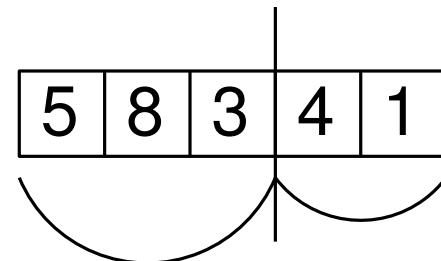
5	8	3	4	1
---	---	---	---	---



Parallel Prefix Sums

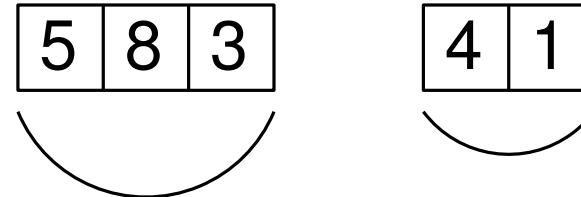


Parallel Prefix Sums



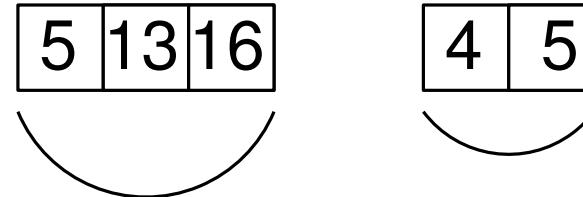
Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



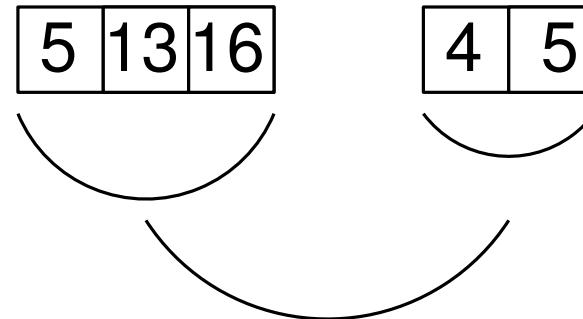
Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



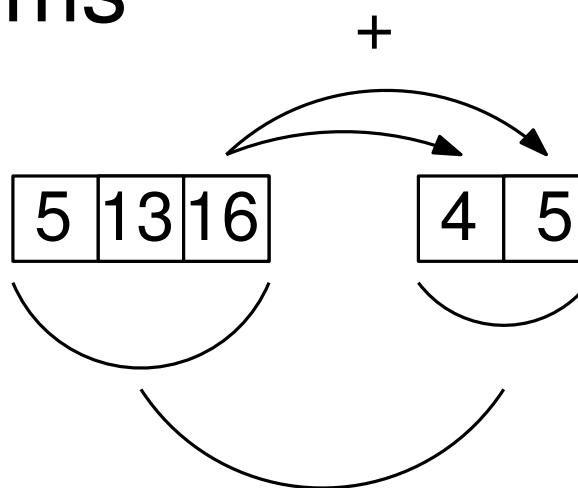
Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



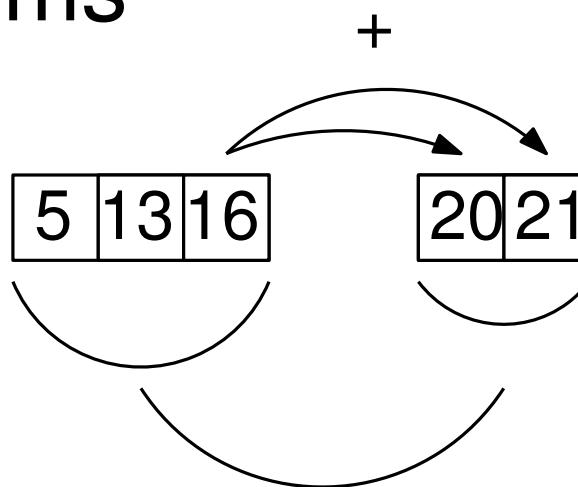
Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



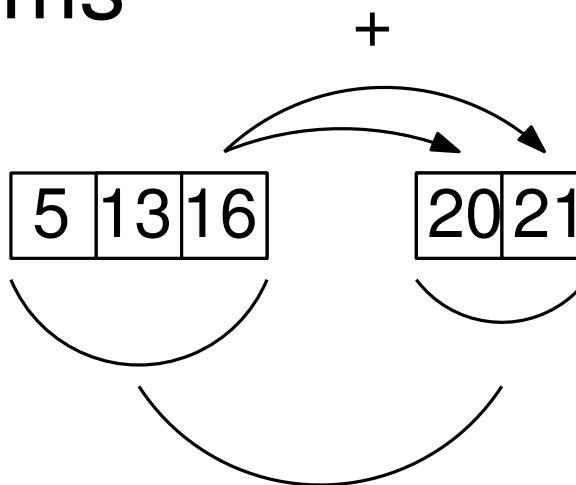
Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

▷ Base case

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

PREFIX-SUMS(A, i, mid)

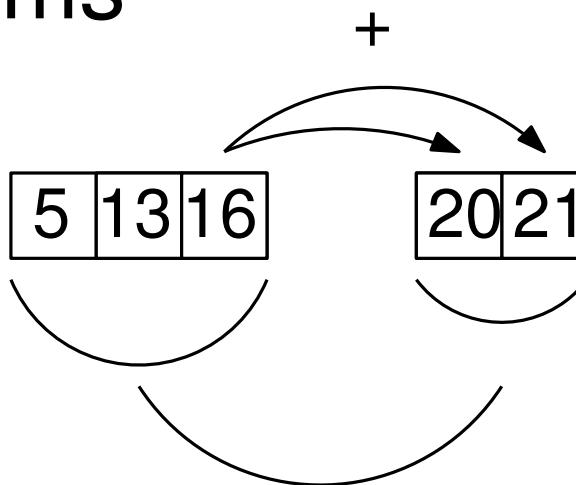
PREFIX-SUMS($A, mid + 1, j$)

for $k = mid + 1$ to j **do**

$$A[k] = A[k] + A[mid]$$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

▷ Base case

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

PREFIX-SUMS(A, i, mid)

PREFIX-SUMS($A, mid + 1, j$)

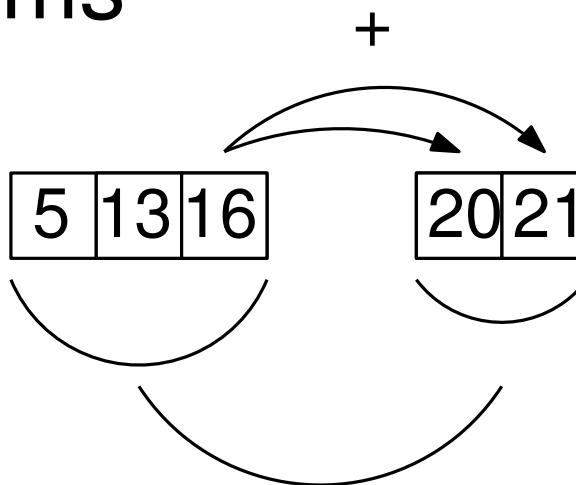
for $k = mid + 1$ to j **do**

$$A[k] = A[k] + A[mid]$$

$$\begin{aligned} W(n) &= 2W(n/2) + O(n) \\ &= O(n \log n) \end{aligned}$$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

▷ Base case

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

PREFIX-SUMS(A, i, mid)

PREFIX-SUMS($A, mid + 1, j$)

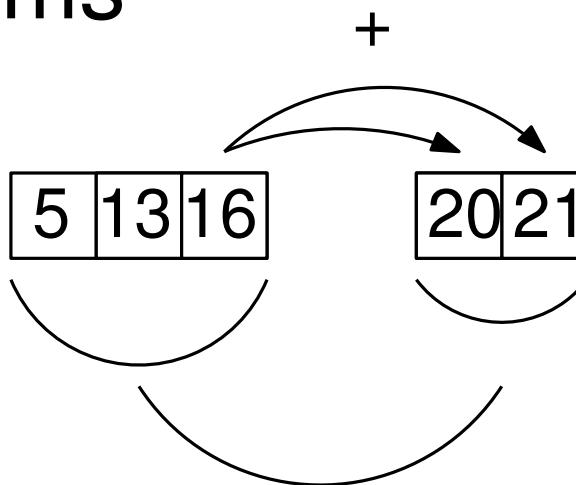
$$\begin{aligned} W(n) &= 2W(n/2) + O(n) \\ &= O(n \log n) \end{aligned}$$

for $k = mid + 1$ to j **in parallel do**

$$A[k] = A[k] + A[mid]$$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

▷ Base case

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

PREFIX-SUMS(A, i, mid)

PREFIX-SUMS($A, mid + 1, j$)

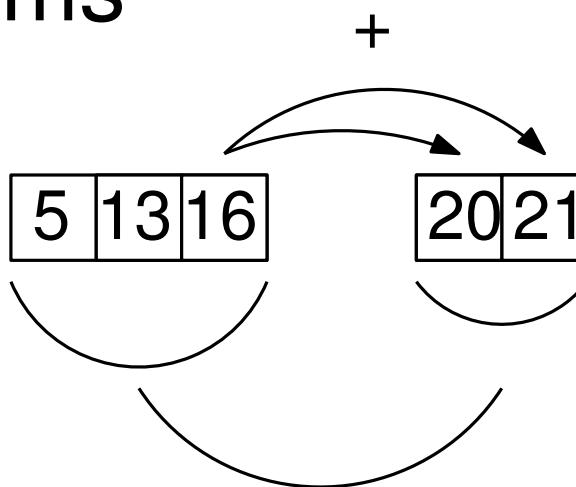
$$\begin{aligned} T(n) &= 2T(n/2) + O(1) \\ &= O(n) \end{aligned}$$

for $k = mid + 1$ to j **in parallel do**

$$A[k] = A[k] + A[mid]$$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

▷ Base case

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

in parallel do

PREFIX-SUMS(A, i, mid)

PREFIX-SUMS($A, mid + 1, j$)

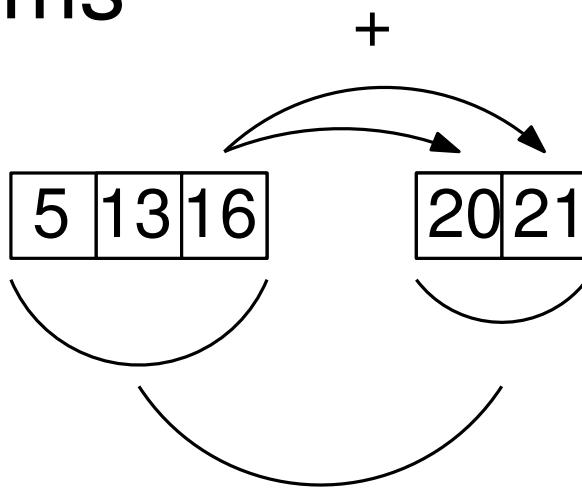
$$\begin{aligned} T(n) &= 2T(n/2) + O(1) \\ &= O(n) \end{aligned}$$

for $k = mid + 1$ to j **in parallel do**

$$A[k] = A[k] + A[mid]$$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

in parallel do

PREFIX-SUMS(A, i, mid)

PREFIX-SUMS($A, mid + 1, j$)

for $k = mid + 1$ to j **in parallel do**

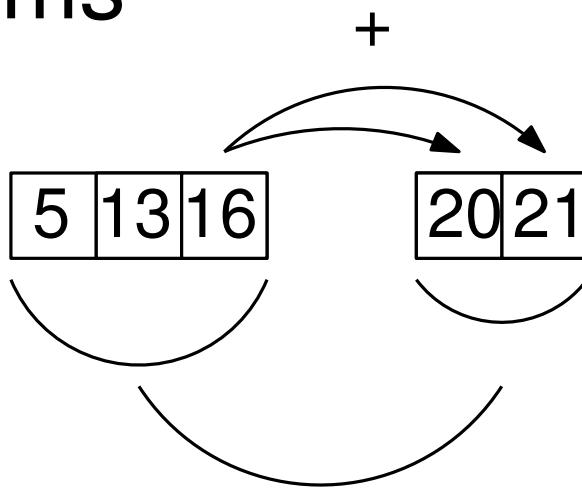
$$A[k] = A[k] + A[mid]$$

$(t_1, t_2) = \text{STARTTWO_THREADS}()$
 t_1 **do:** PREFIX-SUMS(A, i, mid)
 t_2 **do:** PREFIX-SUMS($A, mid + 1, j$)
WAITUNTILFINISHED(t_1, t_2)

$$\begin{aligned} T(n) &= 2T(n/2) + O(1) \\ &= O(n) \end{aligned}$$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

in parallel do

PREFIX-SUMS(A, i, mid)

PREFIX-SUMS($A, mid + 1, j$)

for $k = mid + 1$ to j **in parallel**

$$A[k] = A[k] + A[mid]$$

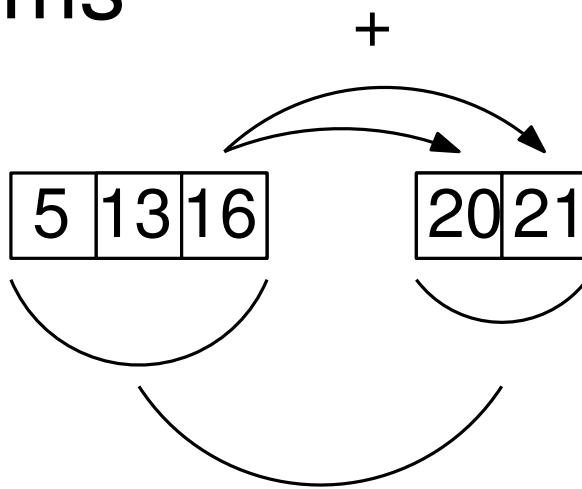
$(t_1, t_2) = \text{STARTTWO_THREADS}()$
 t_1 **do:** PREFIX-SUMS(A, i, mid)
 t_2 **do:** PREFIX-SUMS($A, mid + 1, j$)
WAITUNTILFINISHED(t_1, t_2)

$$T(n) = \max \left\{ T \left(\left\lceil \frac{n}{2} \right\rceil \right), T \left(\left\lfloor \frac{n}{2} \right\rfloor \right) \right\} + O(1)$$

$$\leq T(n/2) + O(1)$$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

in parallel do

PREFIX-SUMS(A, i, mid)

PREFIX-SUMS($A, mid + 1, j$)

for $k = mid + 1$ to j **in parallel**

$$A[k] = A[k] + A[mid]$$

$(t_1, t_2) = \text{STARTTWO_THREADS}()$
 t_1 **do:** PREFIX-SUMS(A, i, mid)
 t_2 **do:** PREFIX-SUMS($A, mid + 1, j$)
WAITUNTILFINISHED(t_1, t_2)

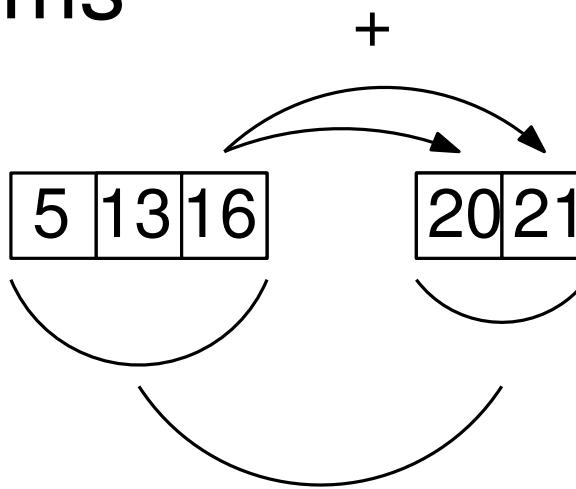
$$T(n) = \max \left\{ T \left(\left\lceil \frac{n}{2} \right\rceil \right), T \left(\left\lfloor \frac{n}{2} \right\rfloor \right) \right\} + O(1)$$

$$\leq T(n/2) + O(1)$$

$$= O(\log n)$$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



function PREFIX-SUMS(A, i, j)

if $i \geq j$ **then return**

$$mid = \left\lfloor \frac{i+j}{2} \right\rfloor$$

in parallel do

PREFIX-SUMS(A, i, mid)

PREFIX-SUMS($A, mid + 1, j$)

for $k = mid + 1$ to j **in parallel do**

$$A[k] = A[k] + A[mid]$$

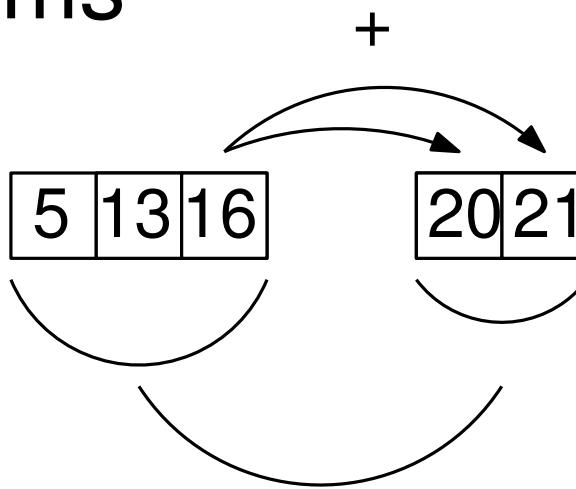
$(t_1, t_2) = \text{STARTTWO_THREADS}()$
 t_1 **do:** PREFIX-SUMS(A, i, mid)
 t_2 **do:** PREFIX-SUMS($A, mid + 1, j$)
WAITUNTILFINISHED(t_1, t_2)

Work: $W(n) = O(n \log n)$

Time: $T(n) = O(\log n)$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



Best sequential time

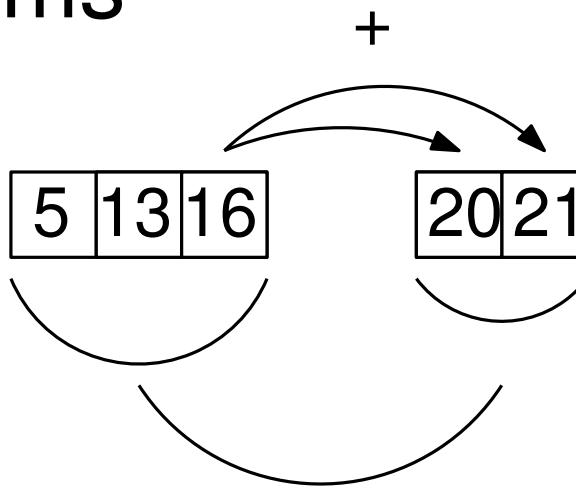
```
function PREFIX-SUMS( $A, i, j$ )
    if  $i \geq j$  then return
     $mid = \lfloor \frac{i+j}{2} \rfloor$ 
    in parallel do
        PREFIX-SUMS( $A, i, mid$ )
        PREFIX-SUMS( $A, mid + 1, j$ )
    for  $k = mid + 1$  to  $j$  in parallel do
         $A[k] = A[k] + A[mid]$ 
```

```
 $(t_1, t_2) = \text{STARTTWO_THREADS}()$ 
 $t_1$  do: PREFIX-SUMS( $A, i, mid$ )
 $t_2$  do: PREFIX-SUMS( $A, mid + 1, j$ )
WAITUNTILFINISHED( $t_1, t_2$ )
```

Work: $W(n) = O(n \log n)$
Time: $T(n) = O(\log n)$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



Best sequential time

$$T(n) = O(n)$$

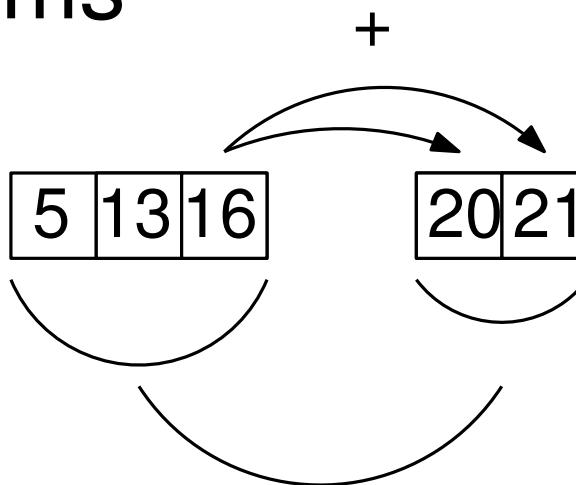
```
function PREFIX-SUMS( $A, i, j$ )
    if  $i \geq j$  then return
     $mid = \lfloor \frac{i+j}{2} \rfloor$ 
    in parallel do
        PREFIX-SUMS( $A, i, mid$ )
        PREFIX-SUMS( $A, mid + 1, j$ )
    for  $k = mid + 1$  to  $j$  in parallel do
         $A[k] = A[k] + A[mid]$ 
```

```
( $t_1, t_2$ ) = STARTTWO_THREADS()
 $t_1$  do: PREFIX-SUMS( $A, i, mid$ )
 $t_2$  do: PREFIX-SUMS( $A, mid + 1, j$ )
WAITUNTILFINISHED( $t_1, t_2$ )
```

Work: $W(n) = O(n \log n)$
Time: $T(n) = O(\log n)$

Parallel Prefix Sums

5	8	3	4	1
---	---	---	---	---



Best sequential time

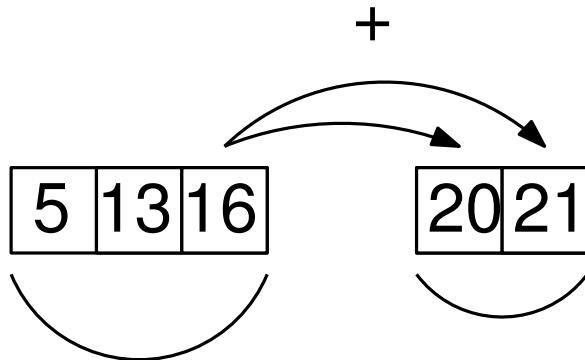
$$T(n) = O(n)$$

```
function PREFIX-SUMS( $A, i, j$ )
    if  $i \geq j$  then return
     $mid = \lfloor \frac{i+j}{2} \rfloor$ 
    in parallel do
        PREFIX-SUMS( $A, i, mid$ )
        PREFIX-SUMS( $A, mid + 1, j$ )
    for  $k$  from  $mid + 1$  to  $j$  in parallel do
         $A[k] = A[k] + A[mid]$ 
```

```
( $t_1, t_2$ ) = STARTTWO_THREADS()
 $t_1$  do: PREFIX-SUMS( $A, i, mid$ )
 $t_2$  do: PREFIX-SUMS( $A, mid + 1, j$ )
WAITUNTILFINISHED( $t_1, t_2$ )
```

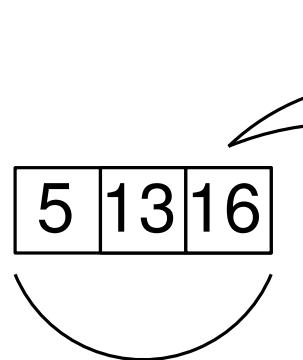
Work: $W(n) = O(n \log n)$
Time: $T(n) = O(\log n)$

Work-efficient Prefix Sums



$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$

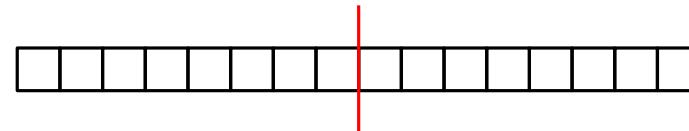
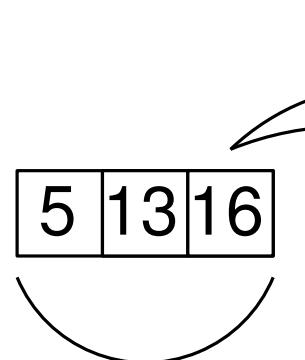
Work-efficient Prefix Sums



$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$

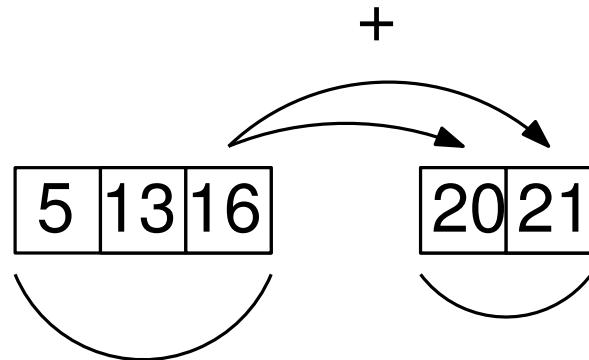


Work-efficient Prefix Sums

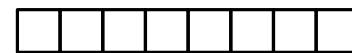
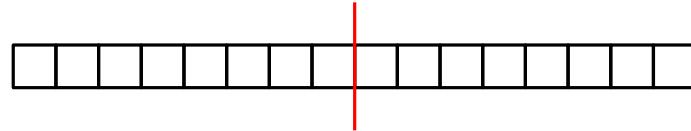


$$\begin{aligned} W(n) &= 2W(n/2) + O(n) \\ &= O(n \log n) \end{aligned}$$

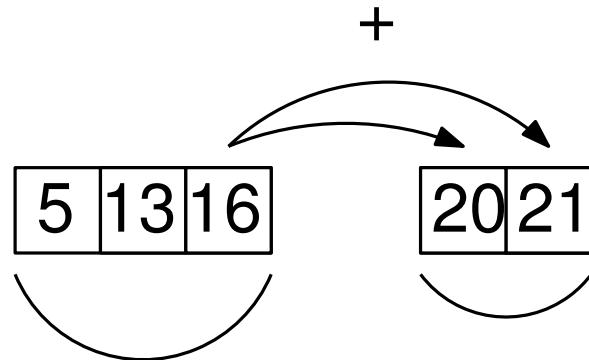
Work-efficient Prefix Sums



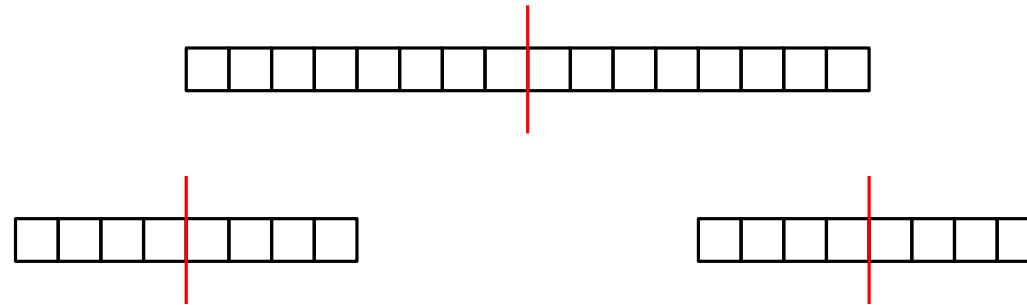
$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



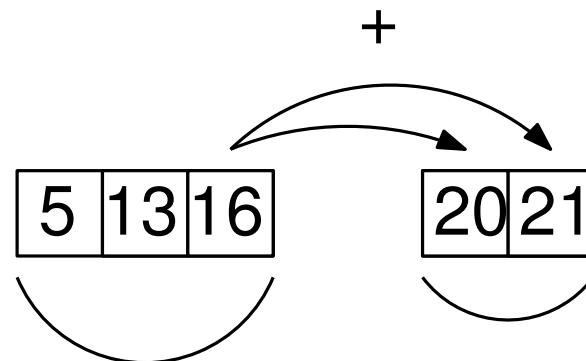
Work-efficient Prefix Sums



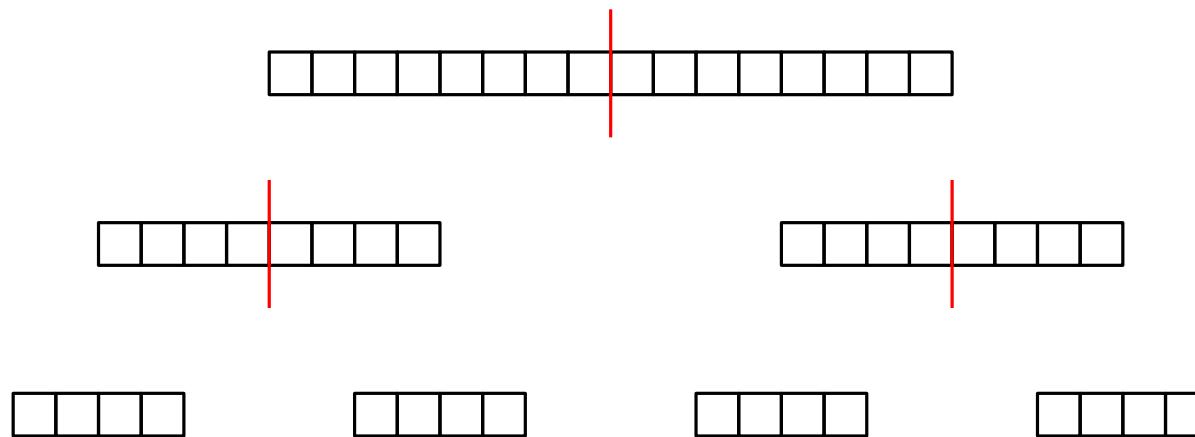
$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



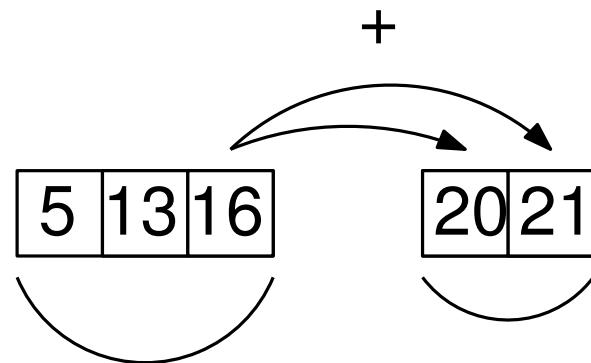
Work-efficient Prefix Sums



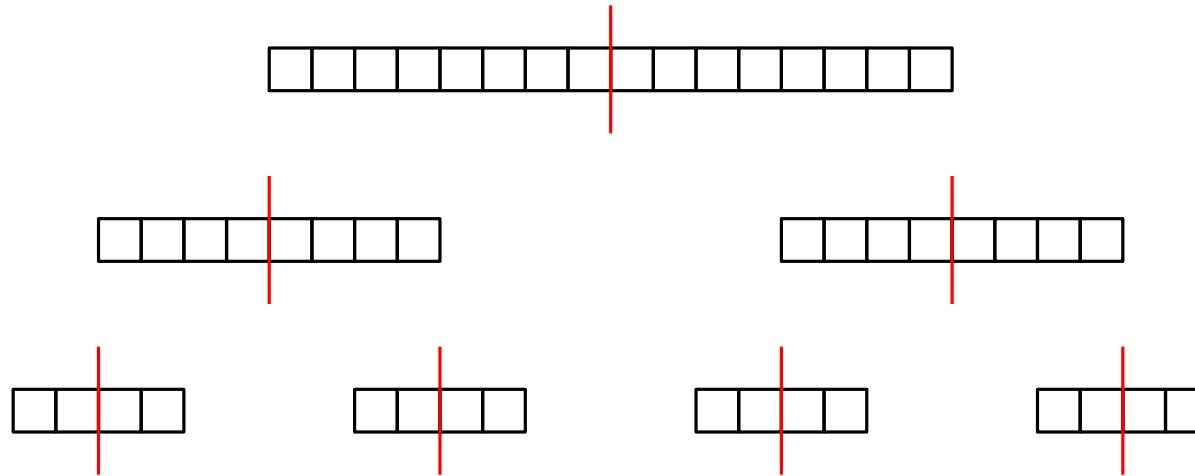
$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



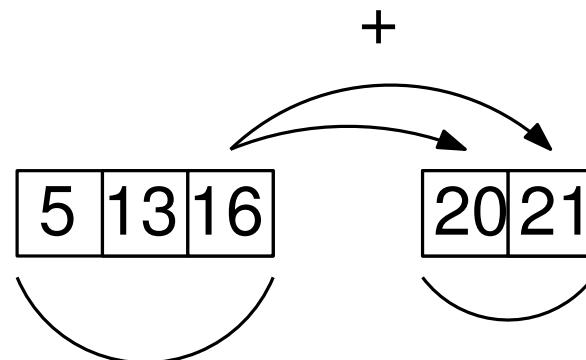
Work-efficient Prefix Sums



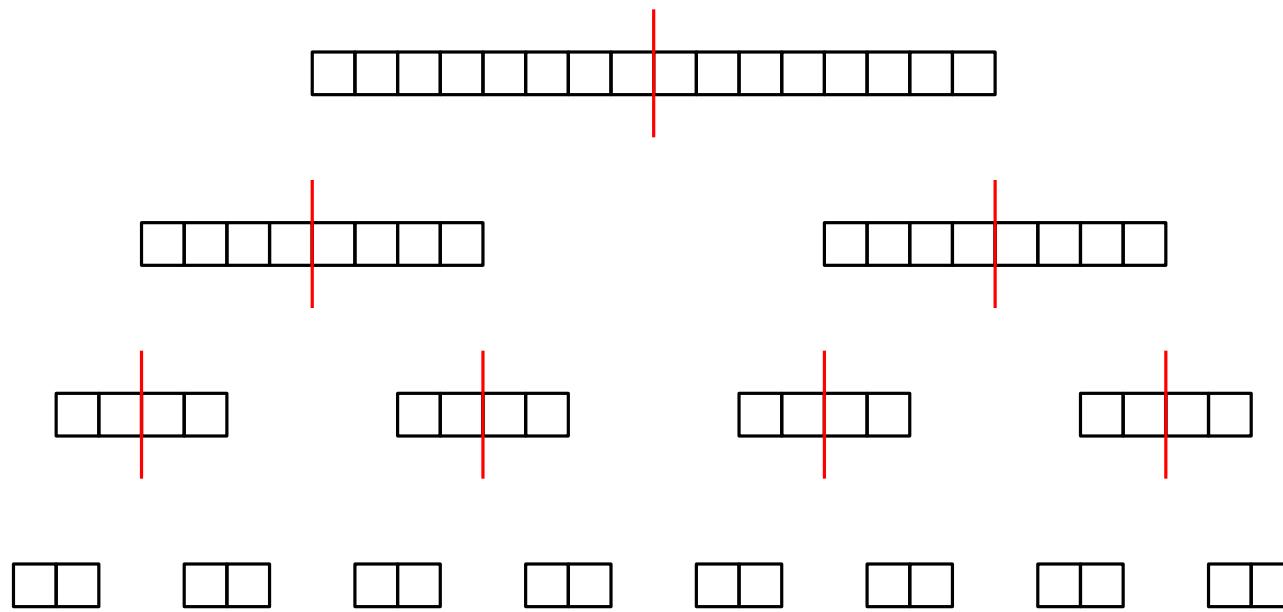
$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



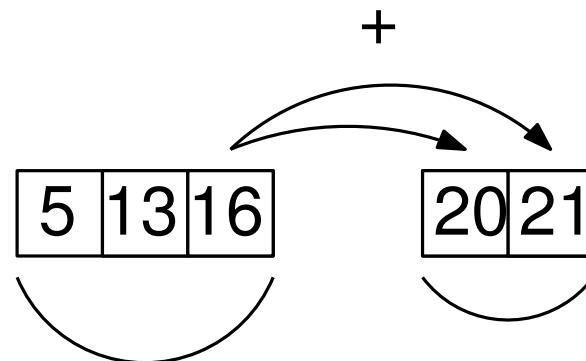
Work-efficient Prefix Sums



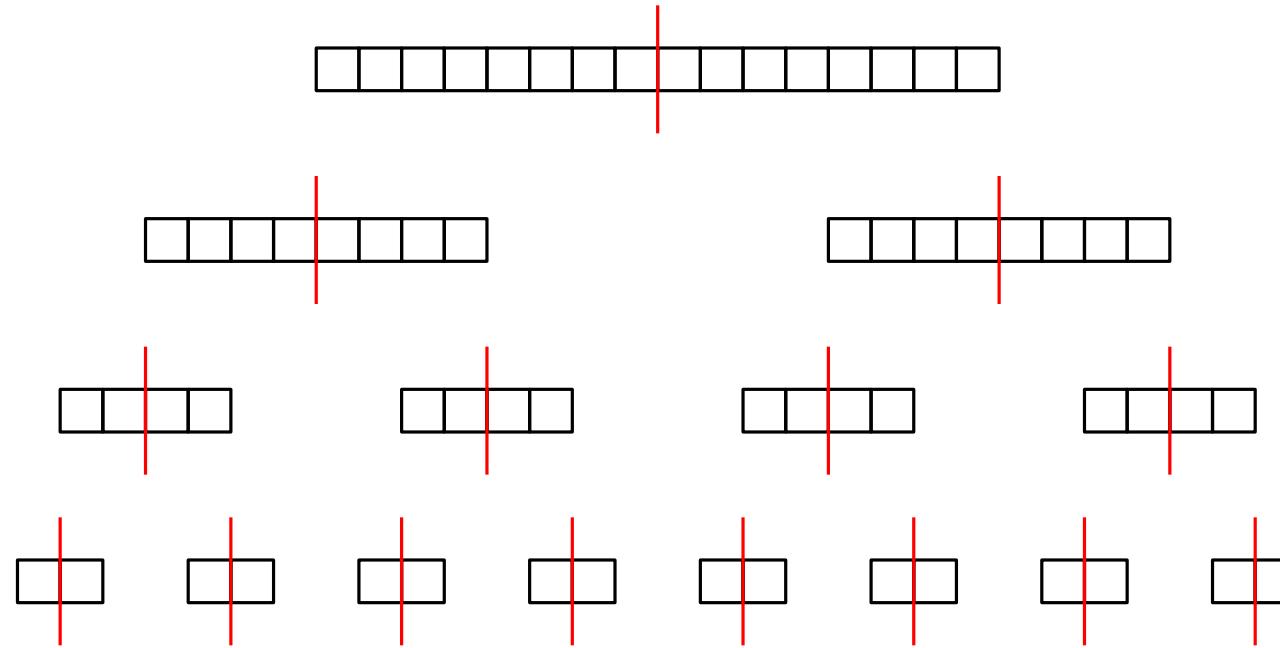
$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



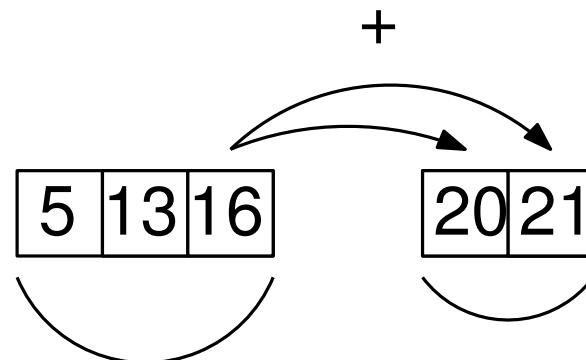
Work-efficient Prefix Sums



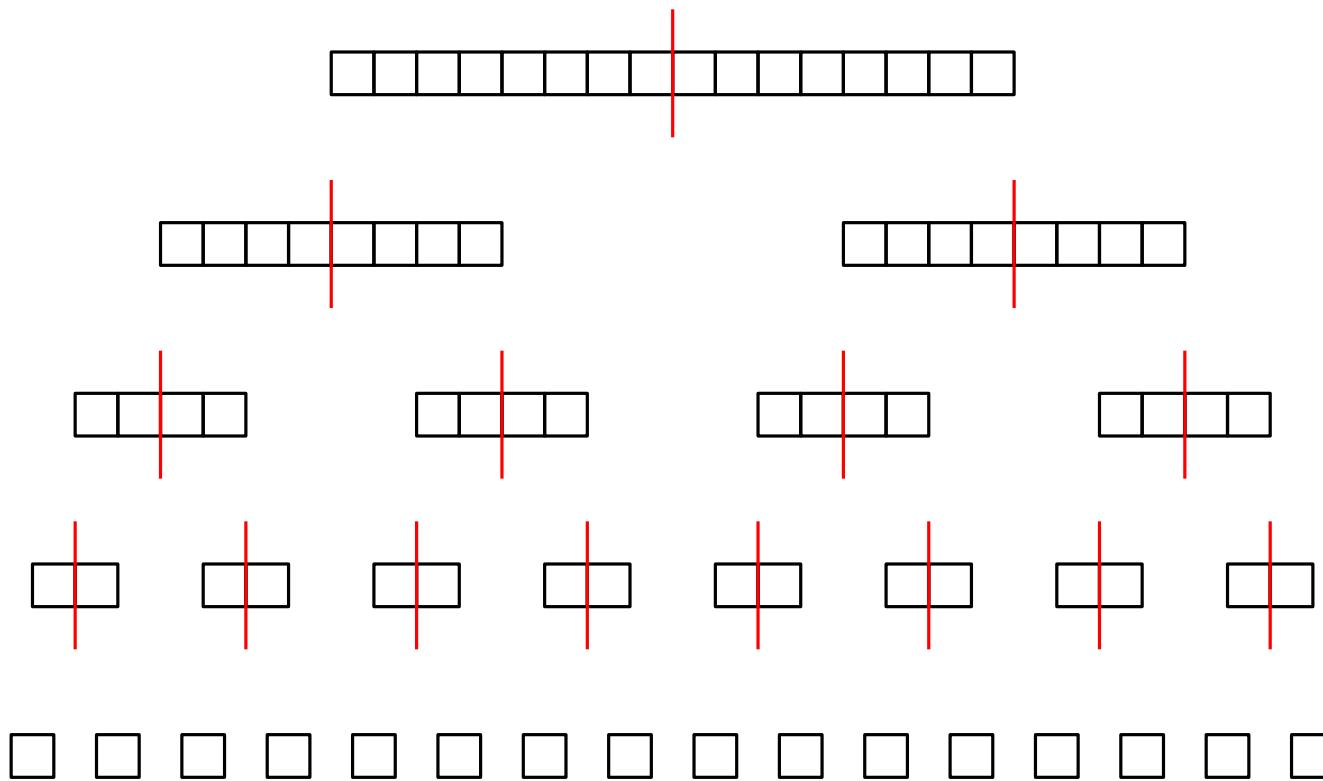
$$\begin{aligned} W(n) &= 2W(n/2) + O(n) \\ &= O(n \log n) \end{aligned}$$



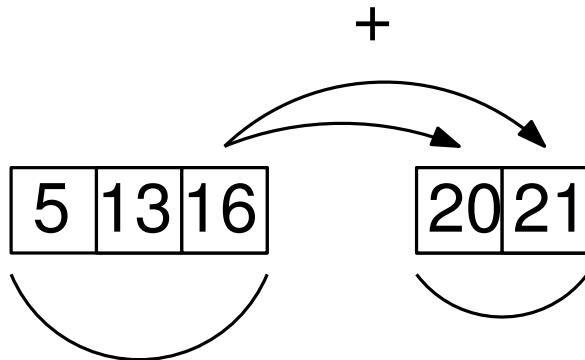
Work-efficient Prefix Sums



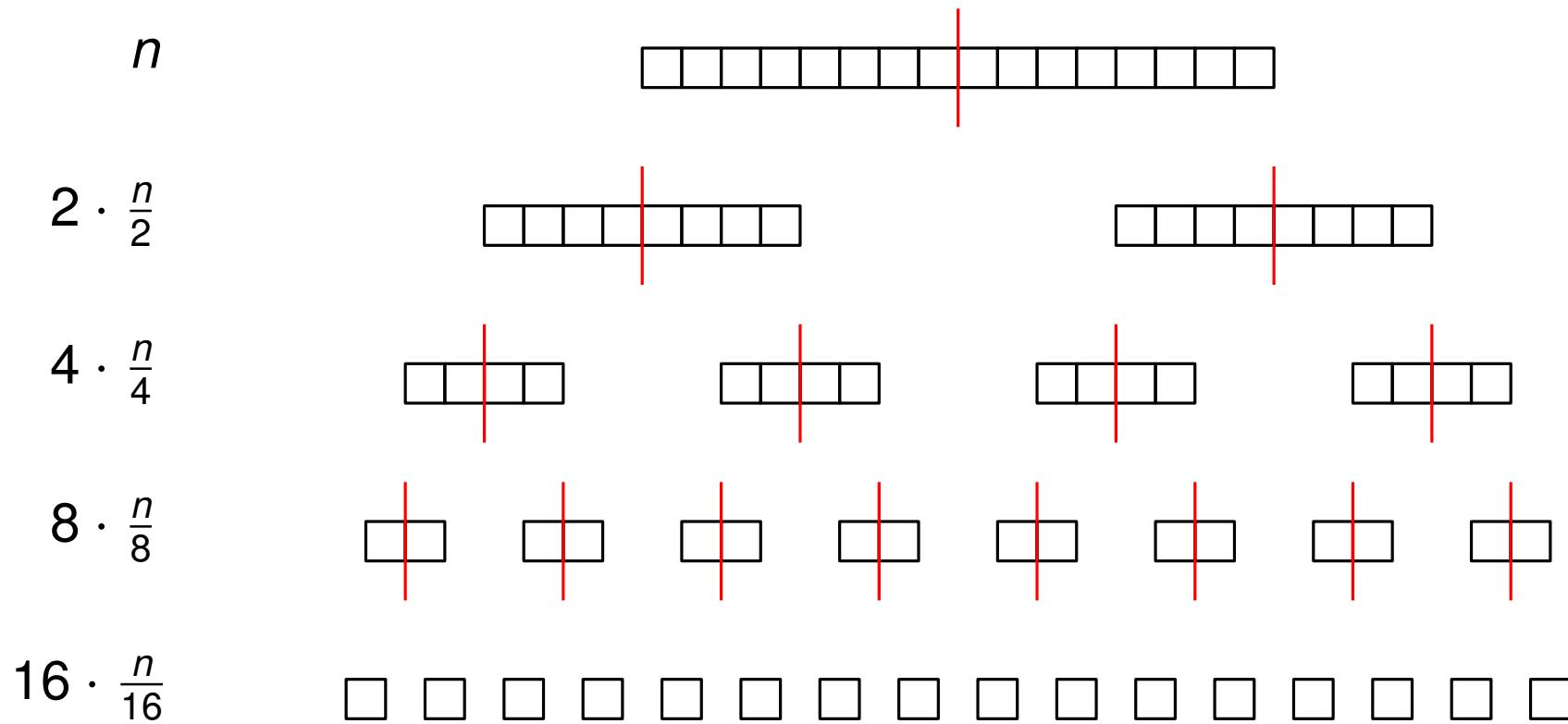
$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



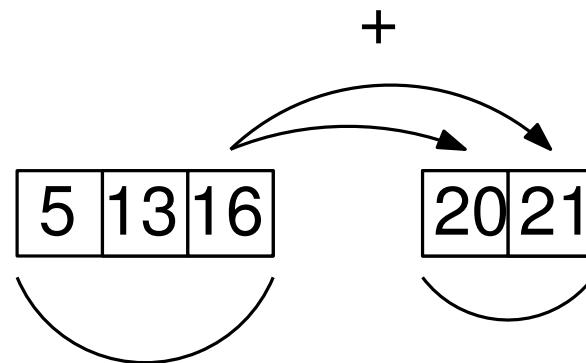
Work-efficient Prefix Sums



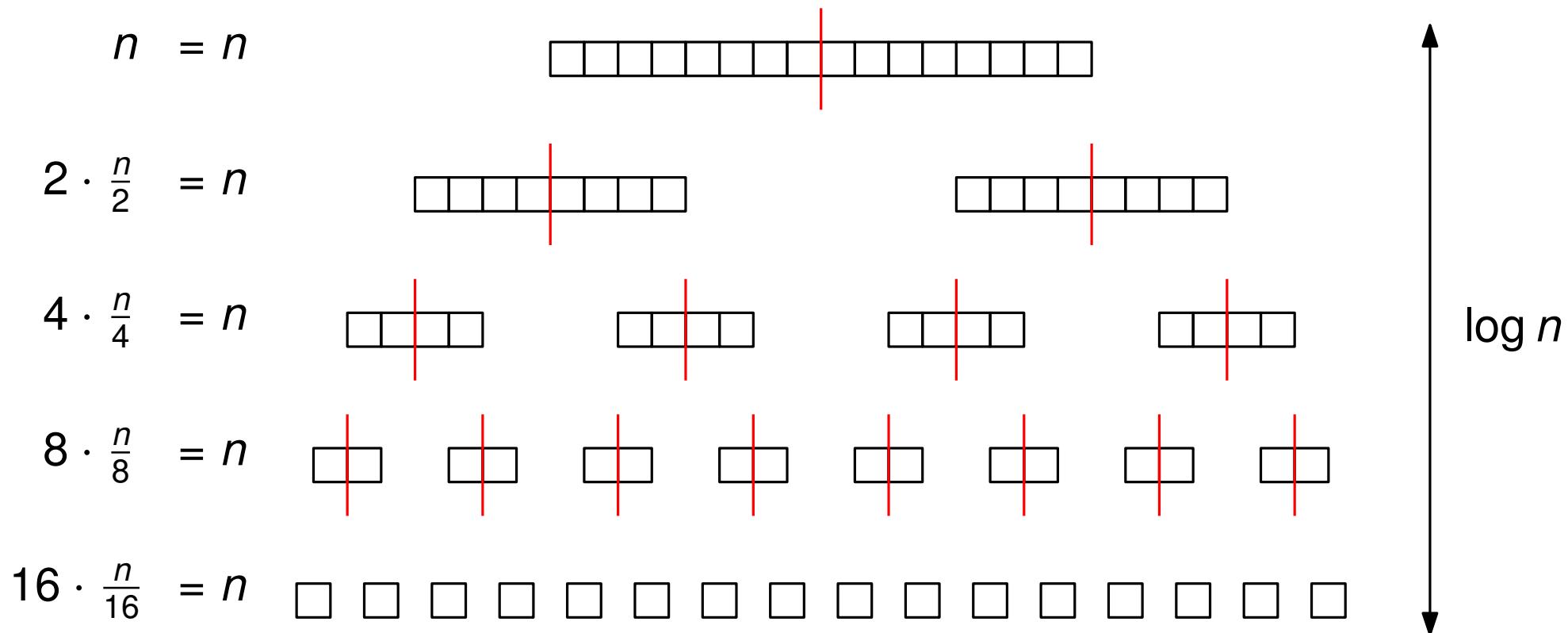
$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



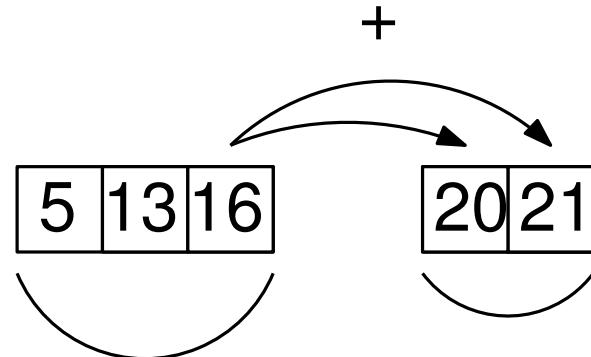
Work-efficient Prefix Sums



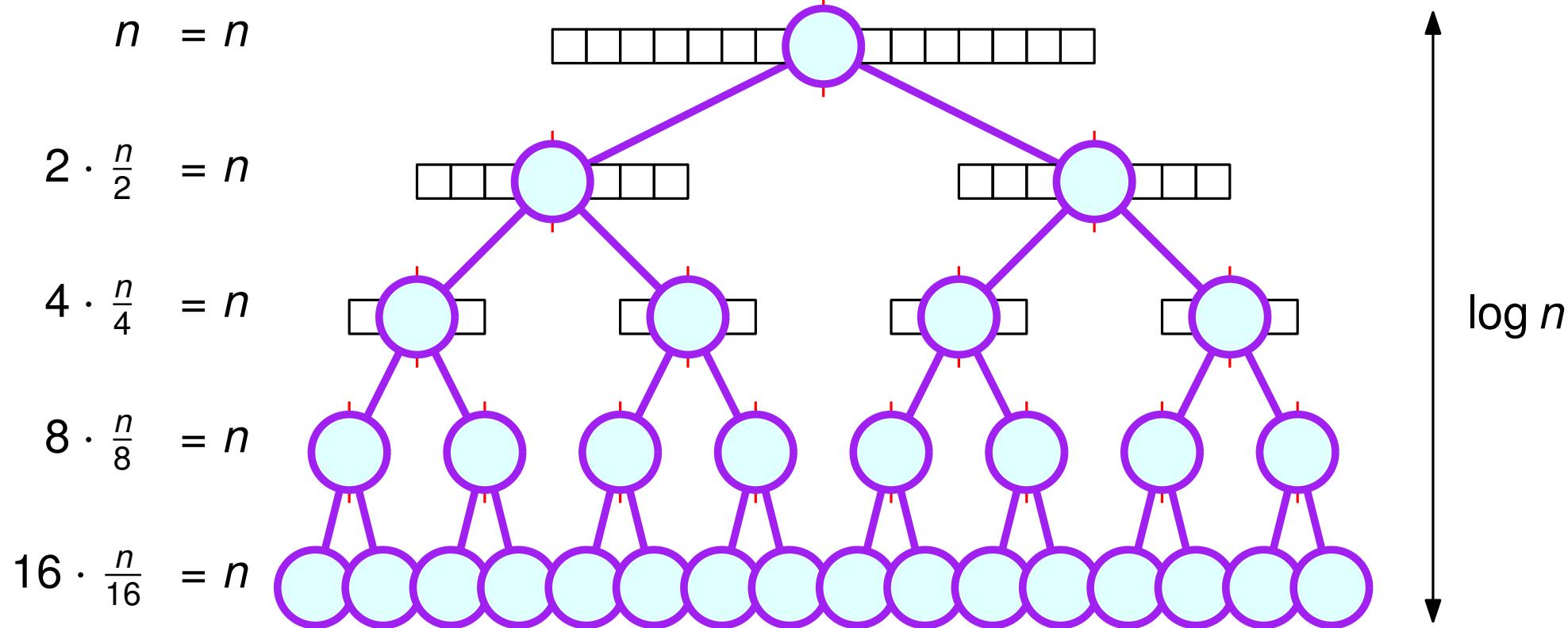
$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



Work-efficient Prefix Sums

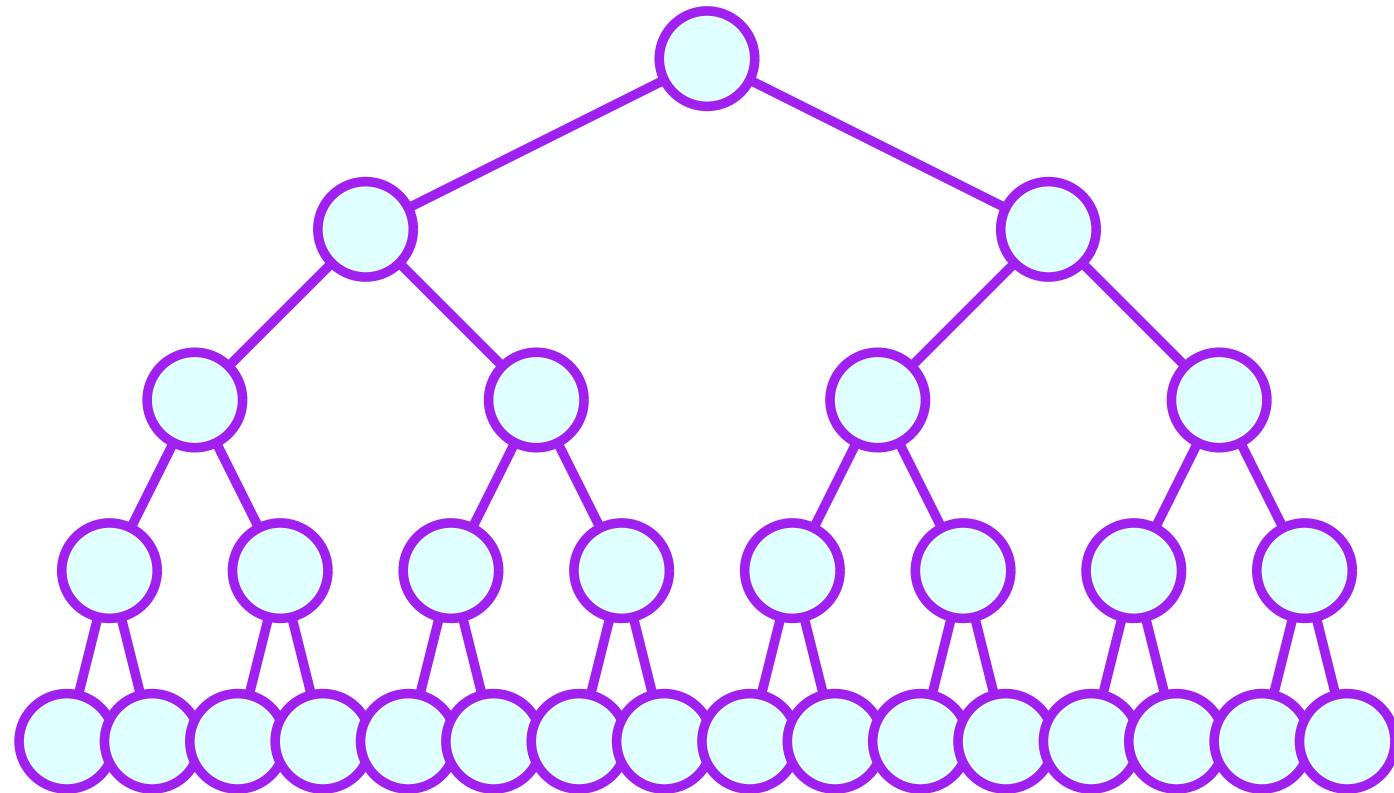


$$\begin{aligned}W(n) &= 2W(n/2) + O(n) \\&= O(n \log n)\end{aligned}$$



Number of Nodes in Perfect Binary Trees

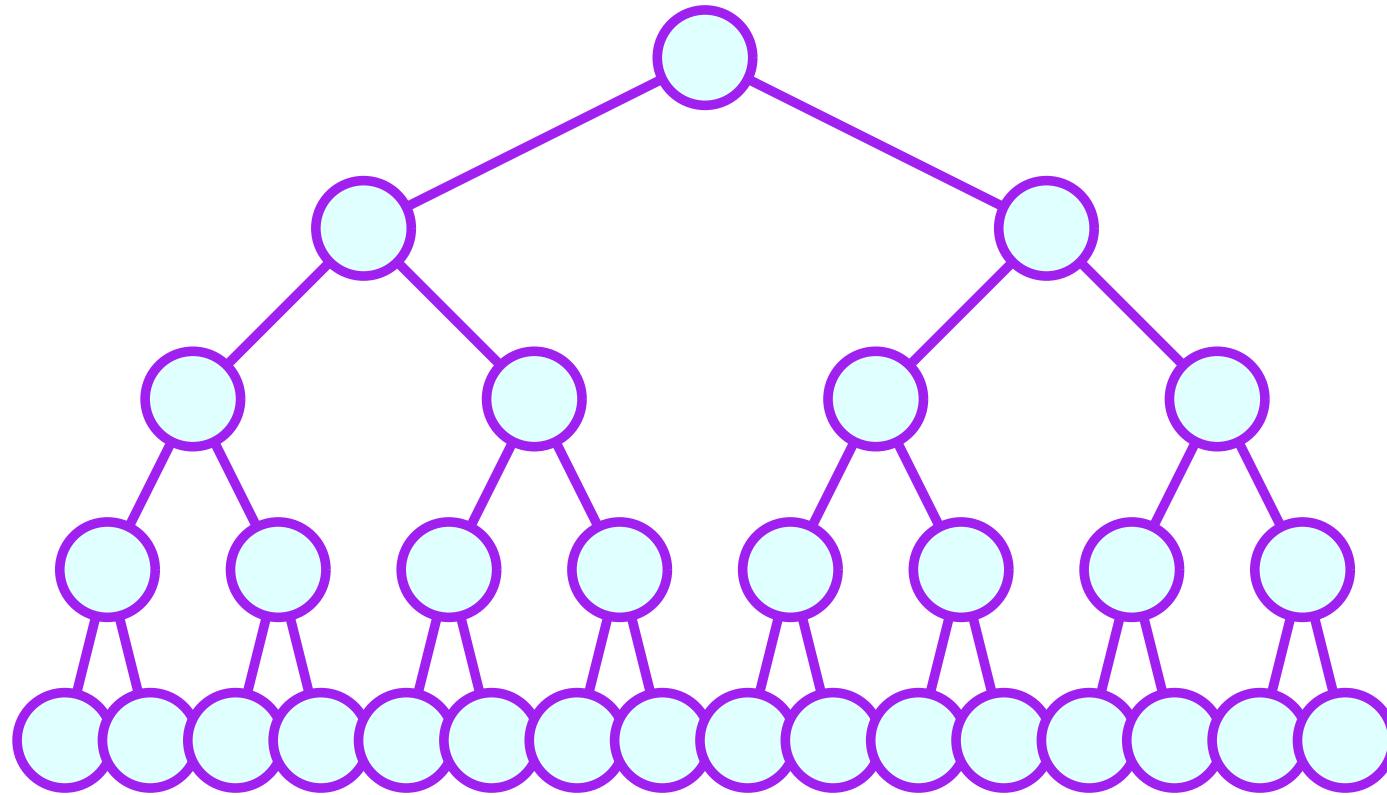
Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*



Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

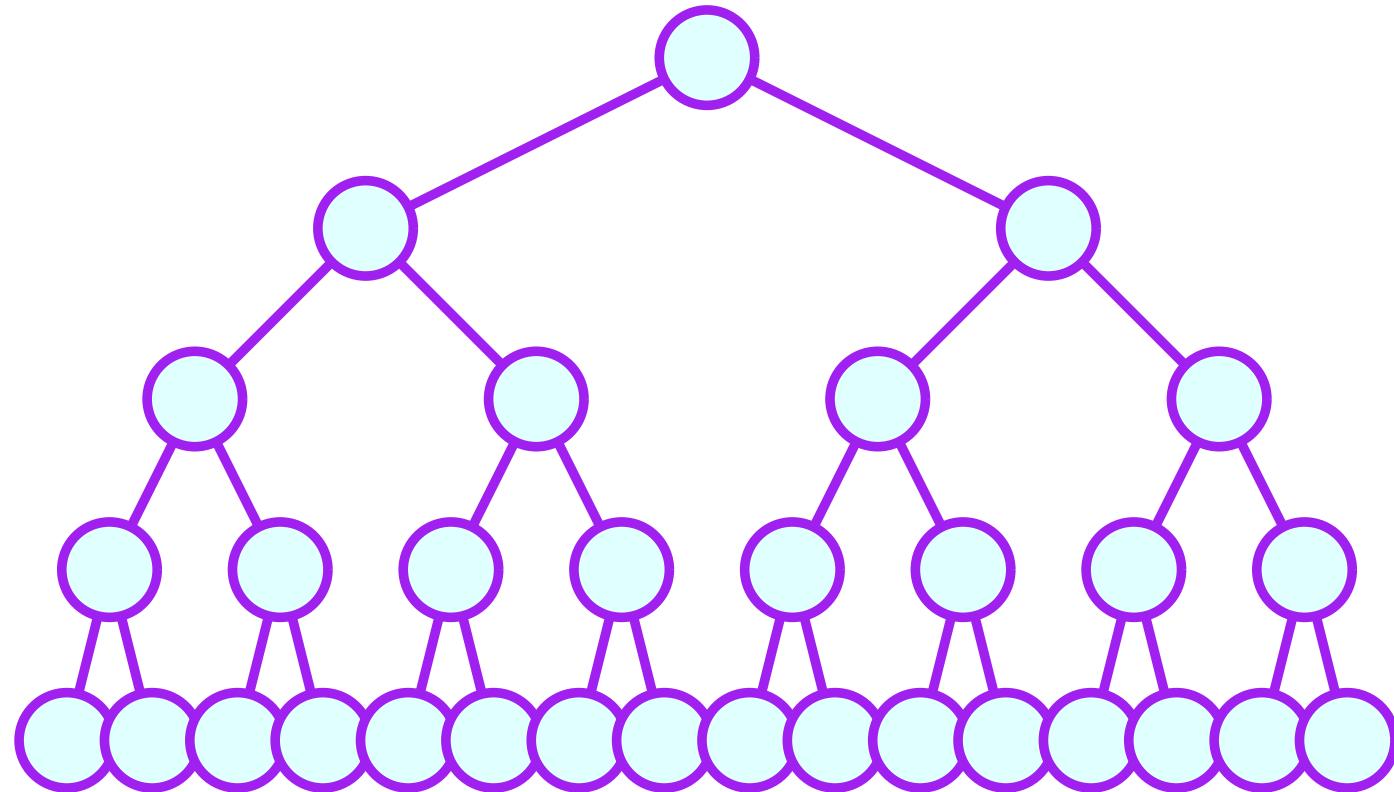


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

Let $f(n)$ = total number of nodes in a perfect tree with n leaves

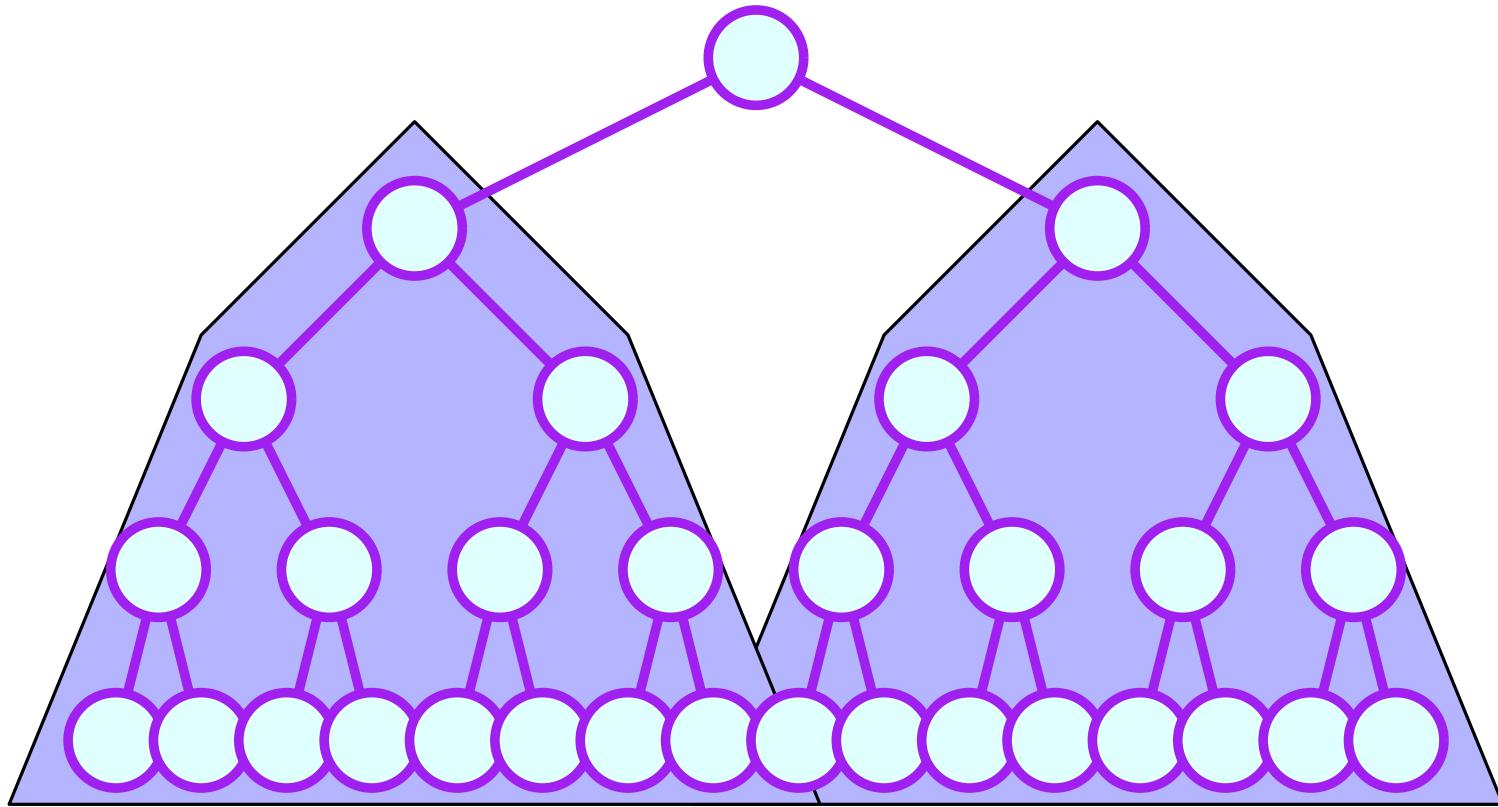


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

Let $f(n)$ = total number of nodes in a perfect tree with n leaves

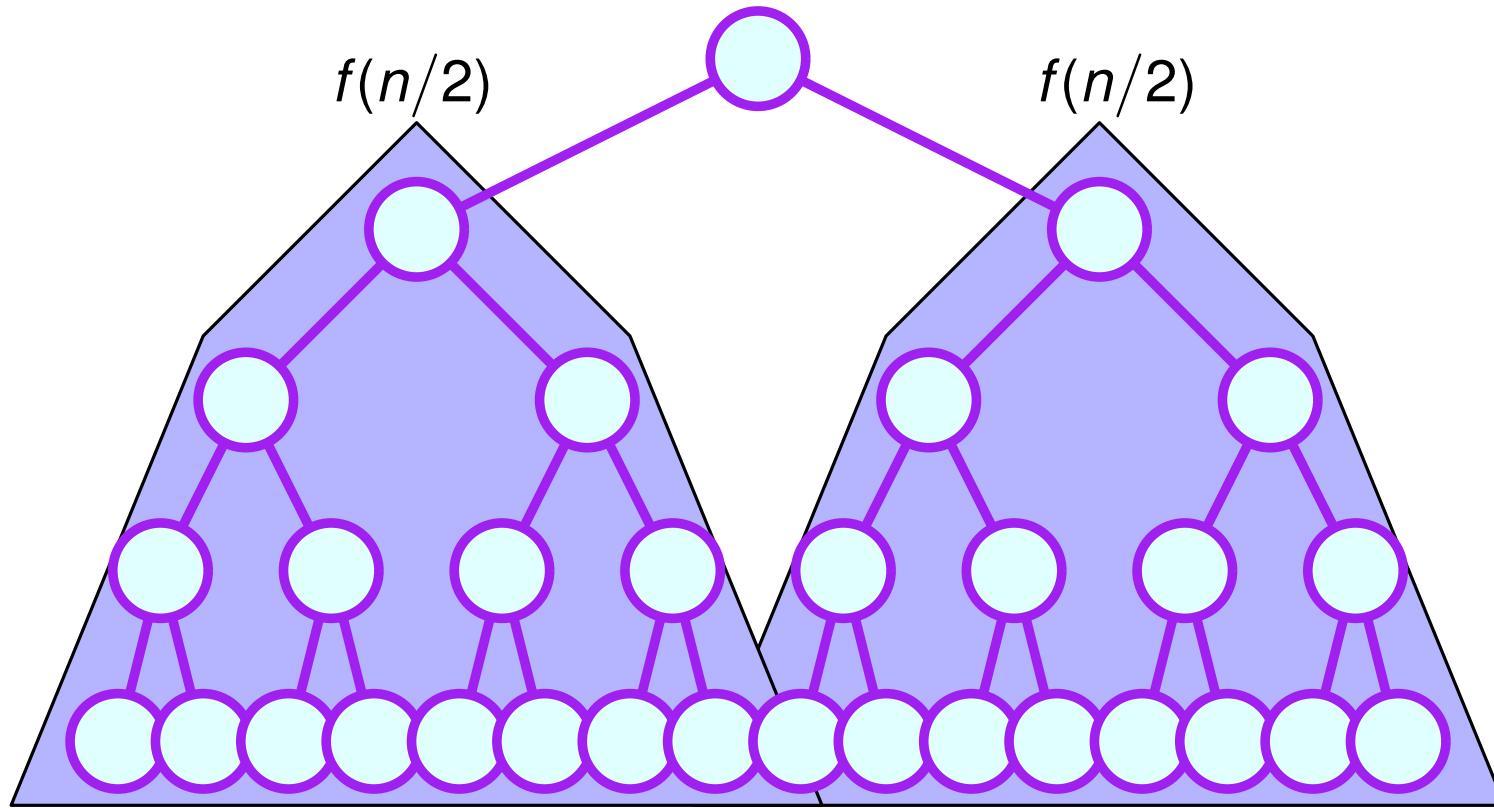


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

Let $f(n)$ = total number of nodes in a perfect tree with n leaves

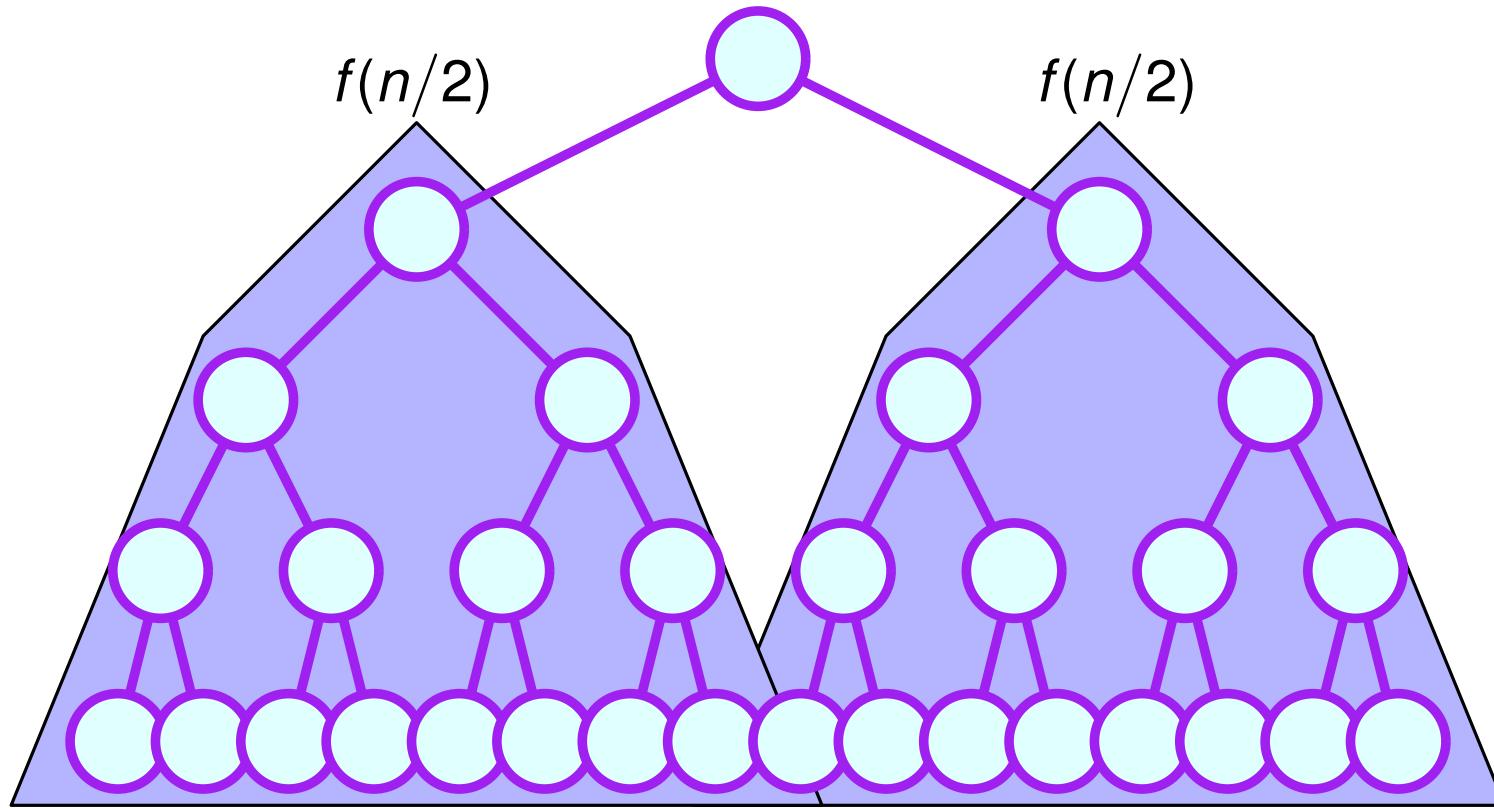


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

$$f(n) = 1 + 2 \cdot f(n/2)$$

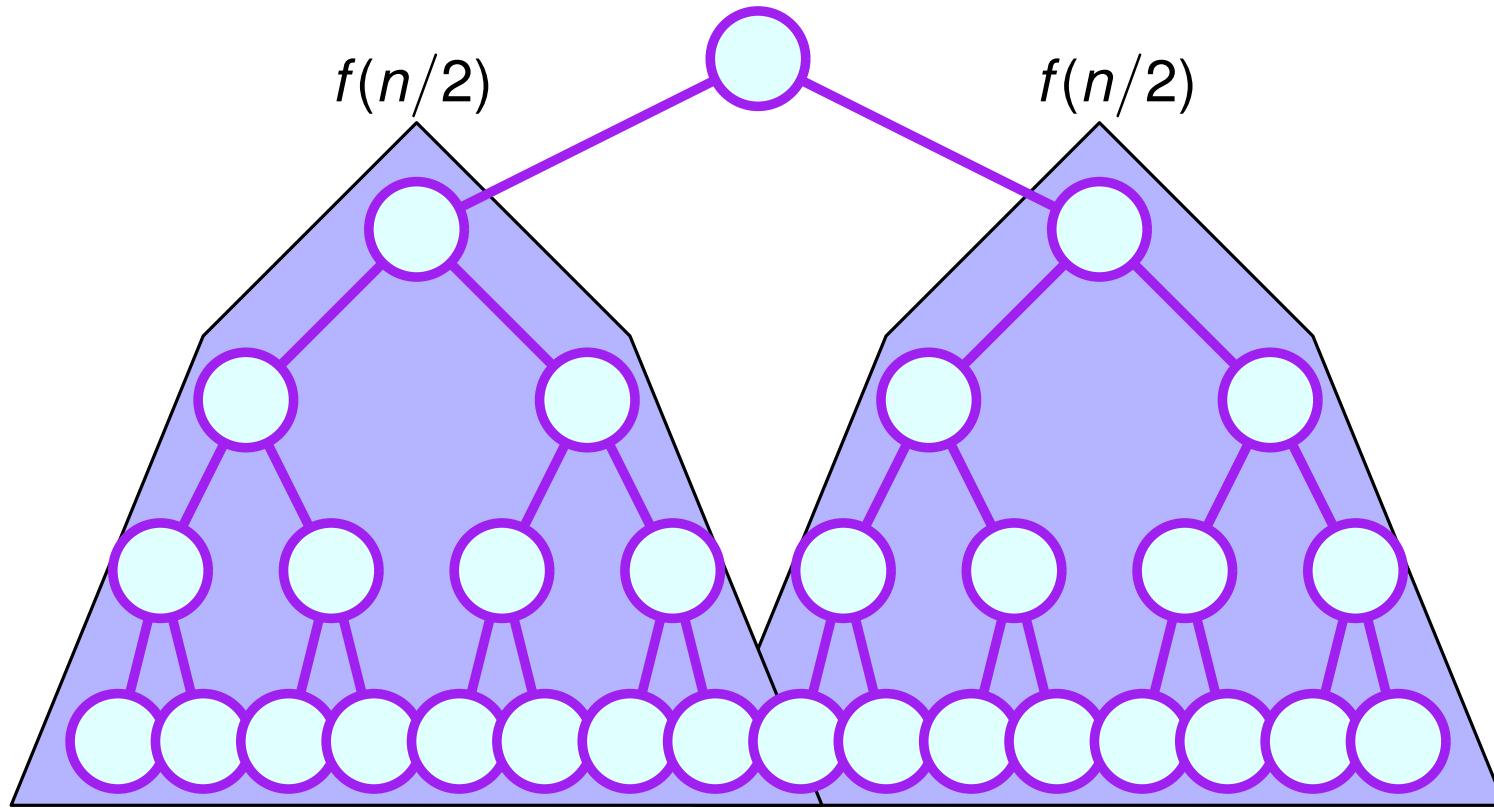


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

$$f(n) = 1 + 2 \cdot f(n/2) = 1 + 2 \cdot \left(2 \cdot \frac{n}{2} - 1\right)$$

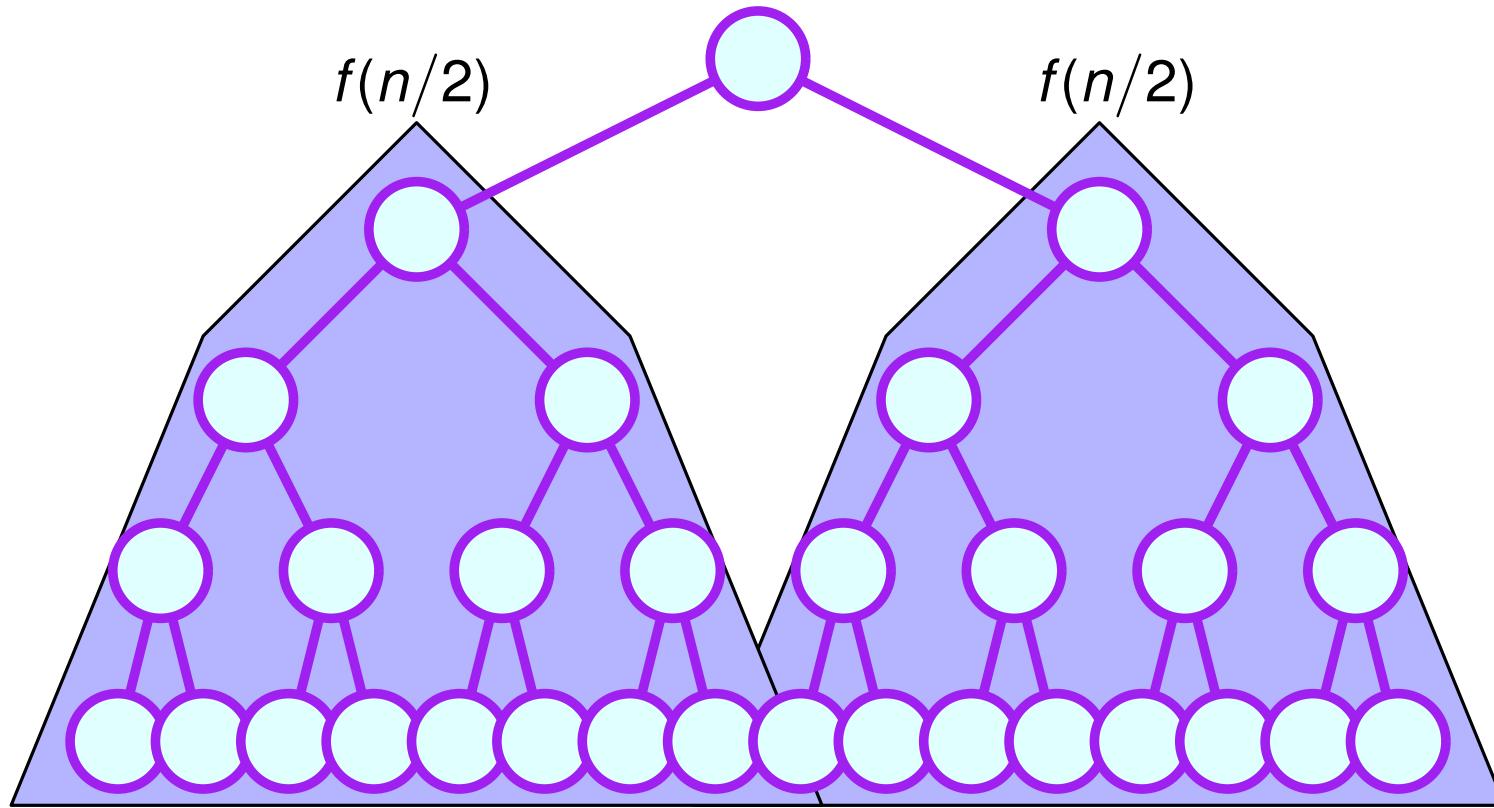


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

$$f(n) = 1 + 2 \cdot f(n/2) = 1 + 2 \cdot \left(2 \cdot \frac{n}{2} - 1\right) = 1 + 2 \cdot (n - 1)$$

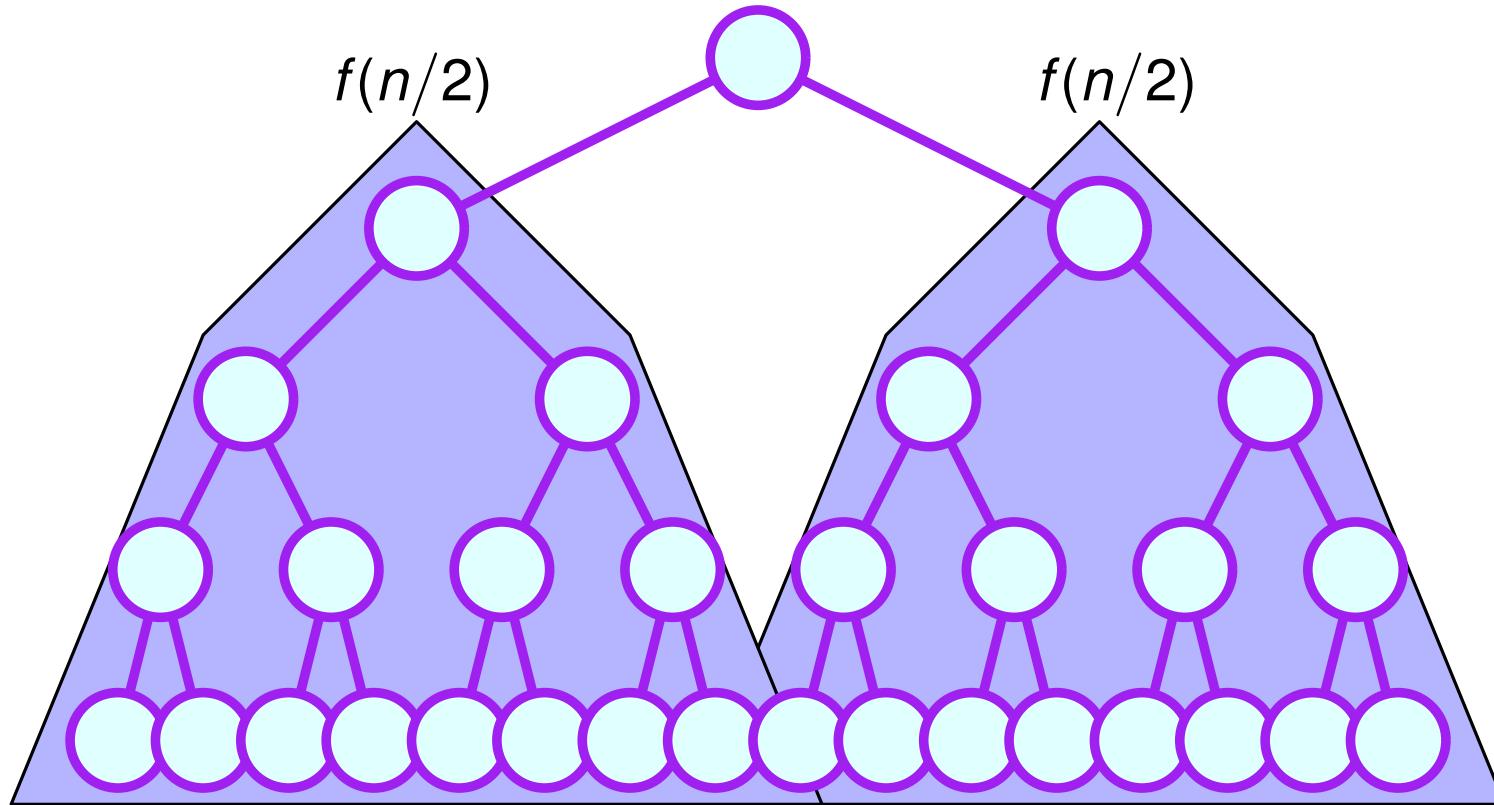


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

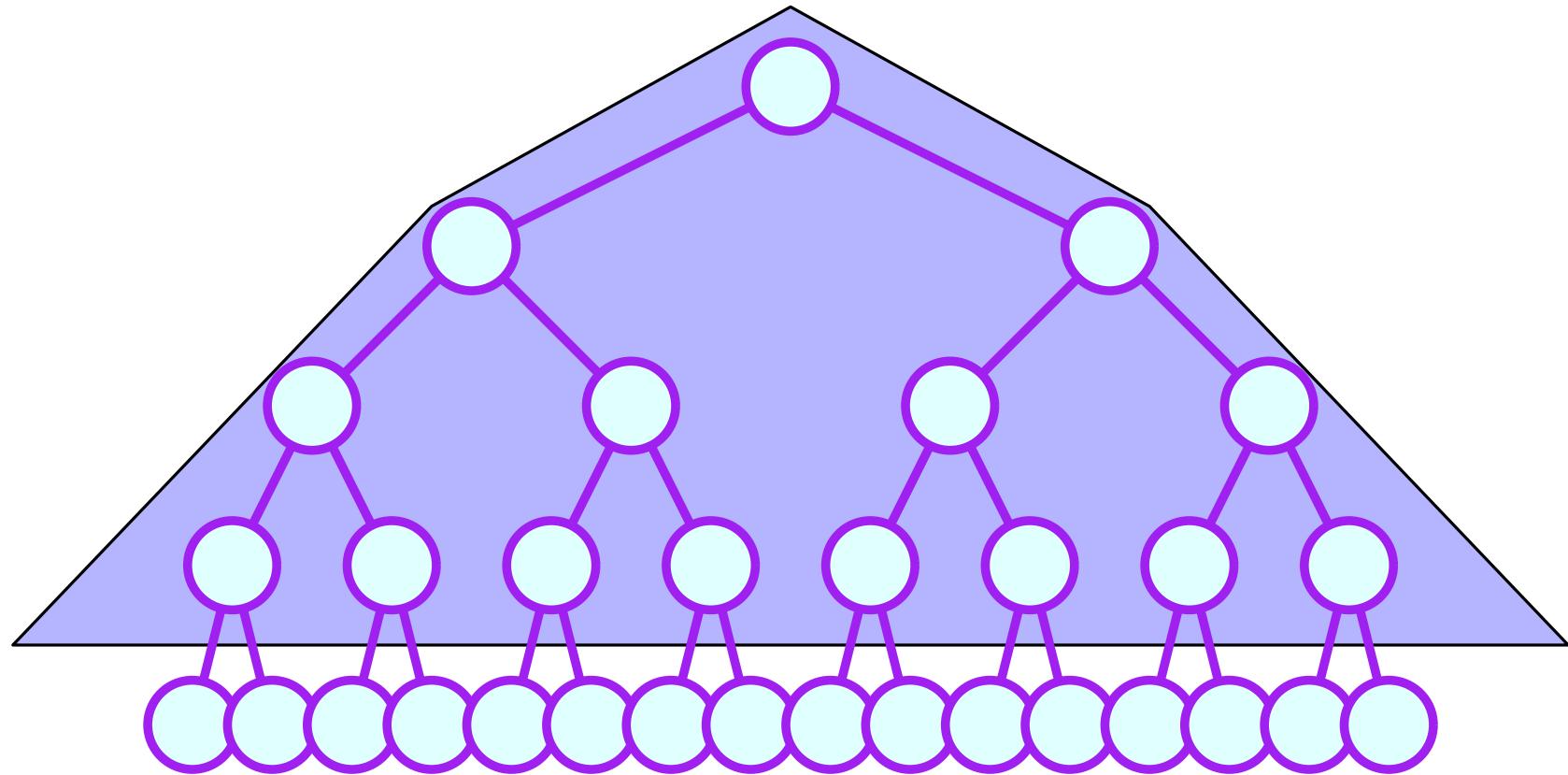
$$f(n) = 1 + 2 \cdot f(n/2) = 1 + 2 \cdot \left(2 \cdot \frac{n}{2} - 1\right) = 1 + 2 \cdot (n - 1) = 1 + (2n - 2) = 2n - 1$$



Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

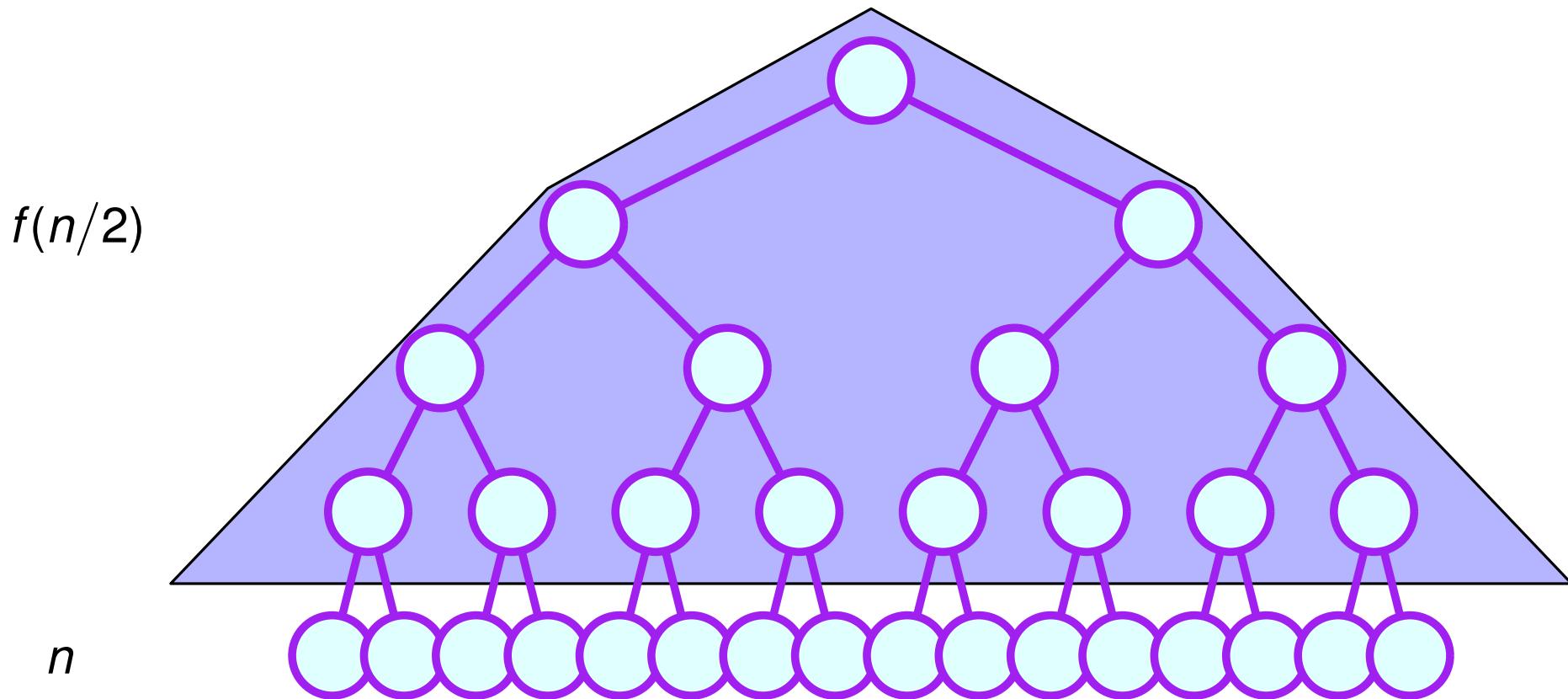
Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.



Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

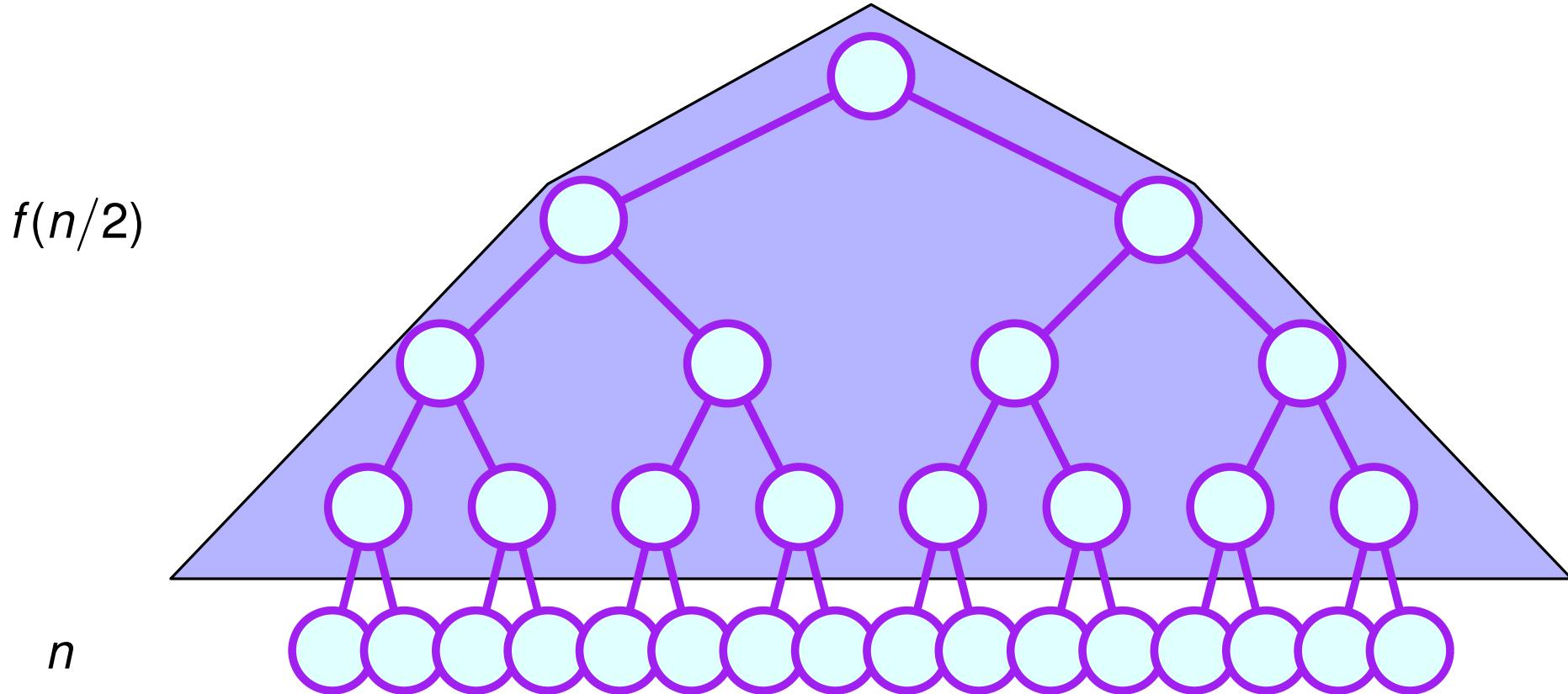


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

$$f(n) = f(n/2) + n$$

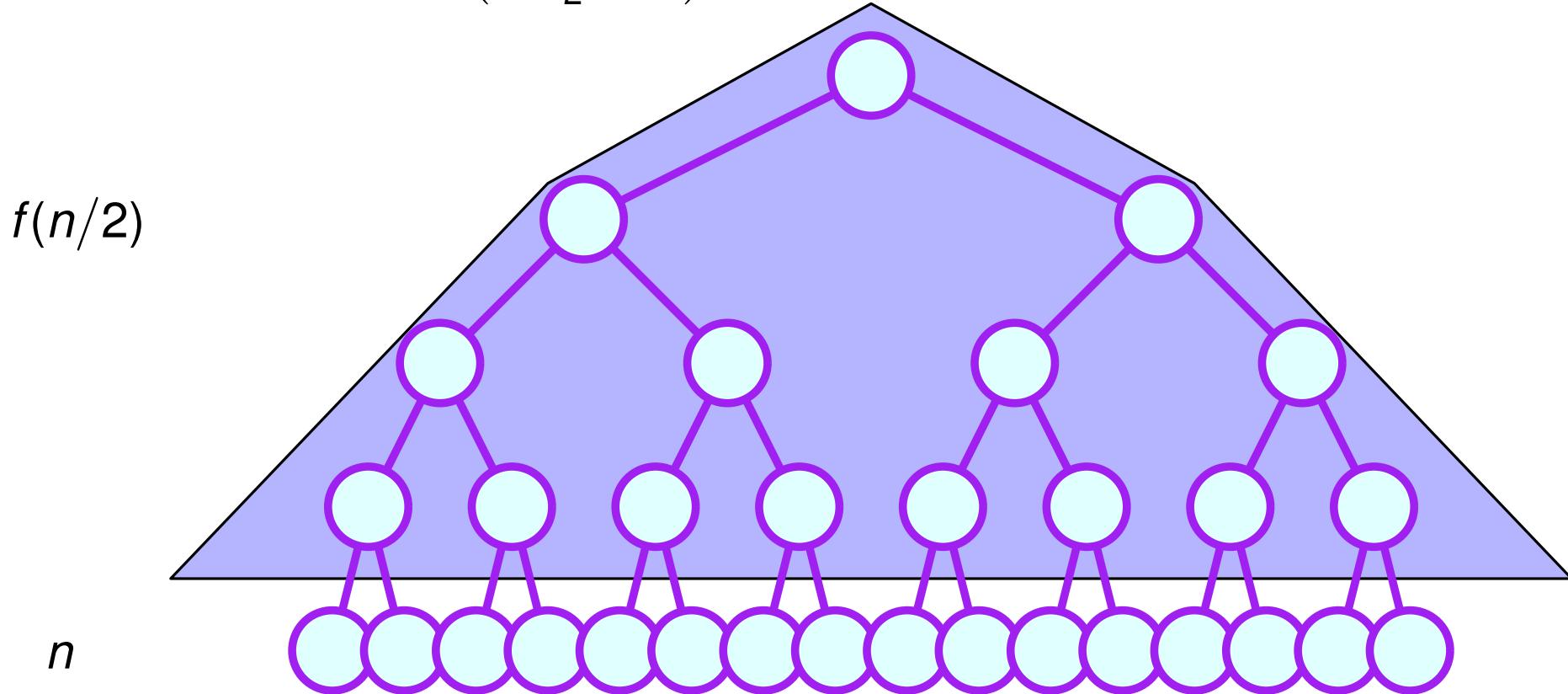


Number of Nodes in Perfect Binary Trees

Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

$$f(n) = f(n/2) + n = \left(2 \cdot \frac{n}{2} - 1\right) + n$$

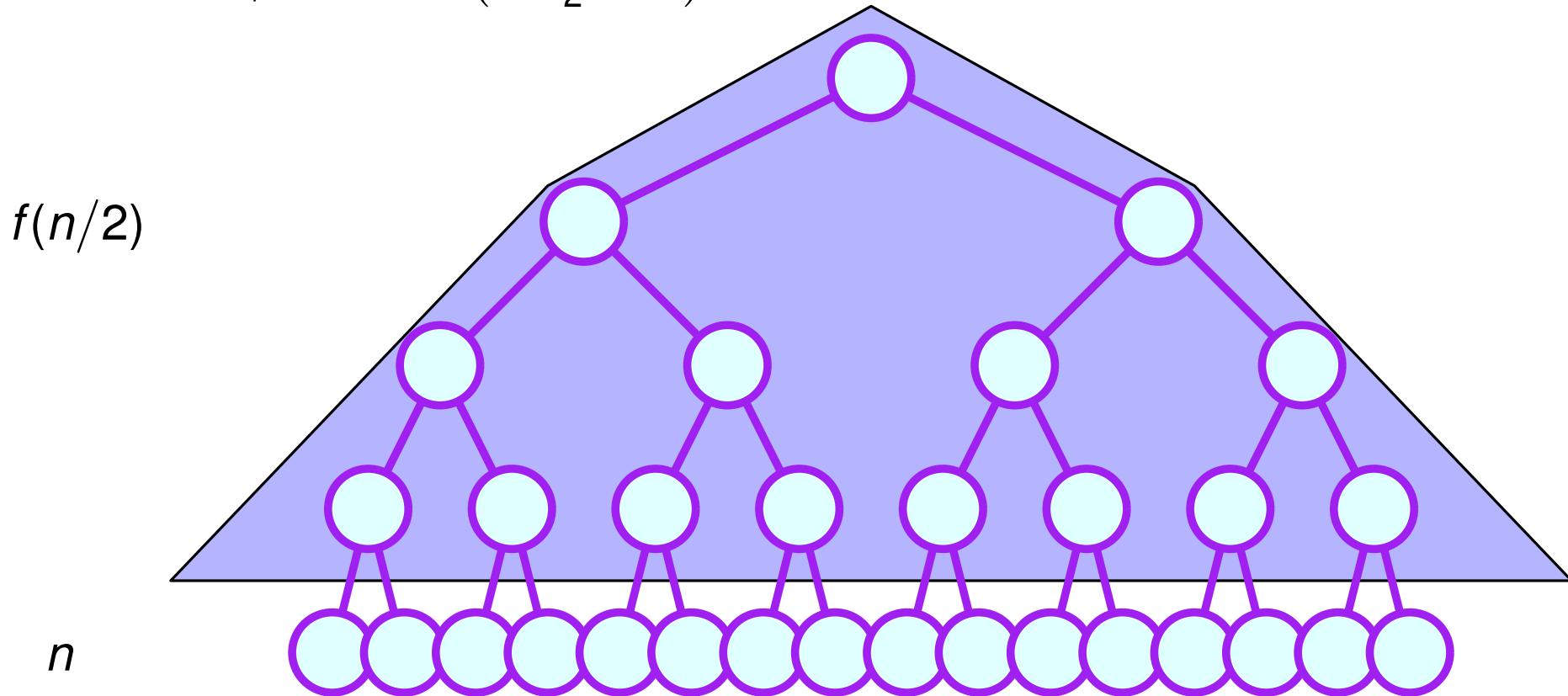


Number of Nodes in Perfect Binary Trees

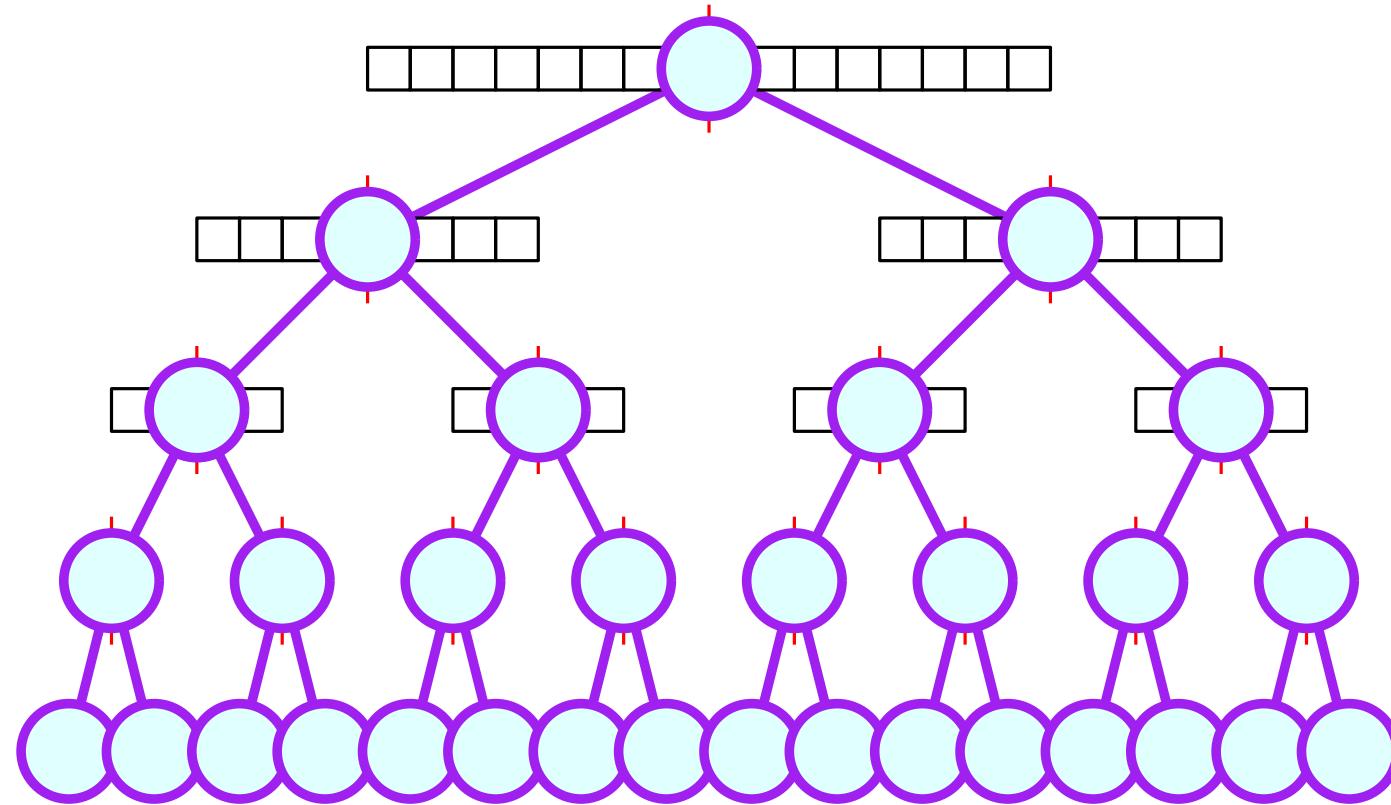
Problem. *Using induction, prove that a perfect binary tree with n leaves has $2n - 1$ total nodes.*

Inductive Hypothesis: For all $k < n$, a perfect binary tree with k leaves has $2k - 1$ total nodes.

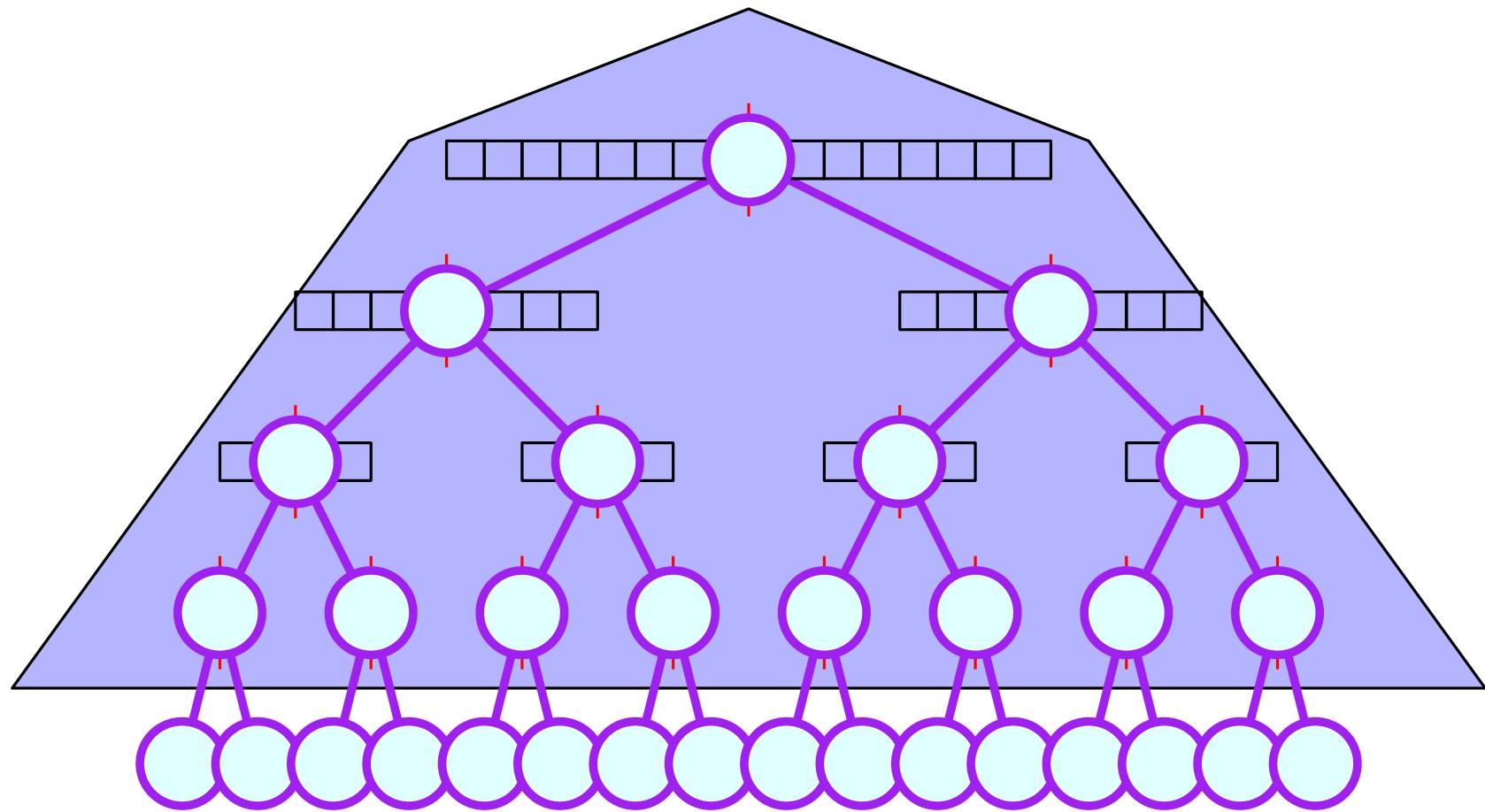
$$f(n) = f(n/2) + n = \left(2 \cdot \frac{n}{2} - 1\right) + n = (n - 1) + n = 2n - 1$$



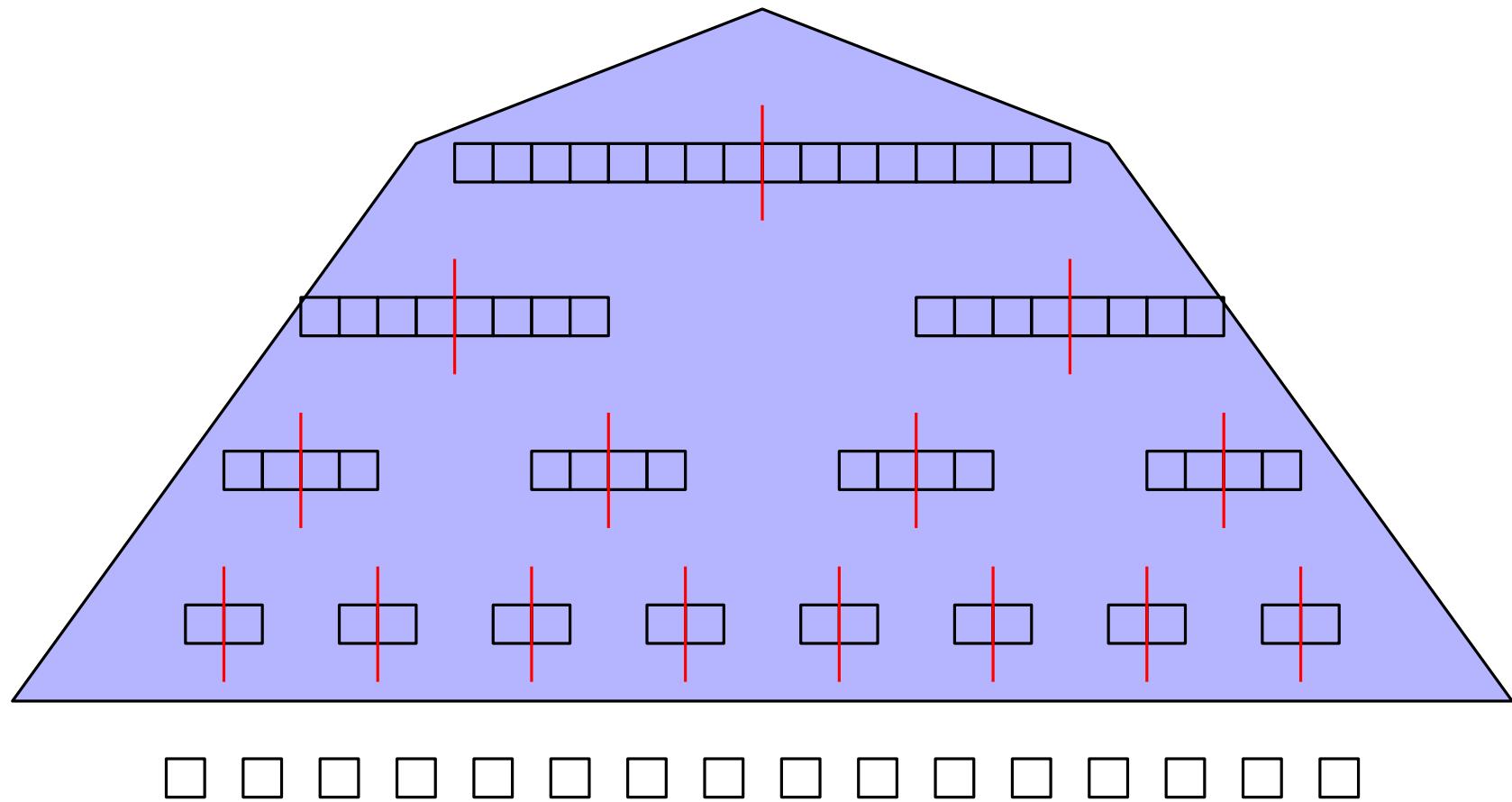
Work-efficient Prefix Sums



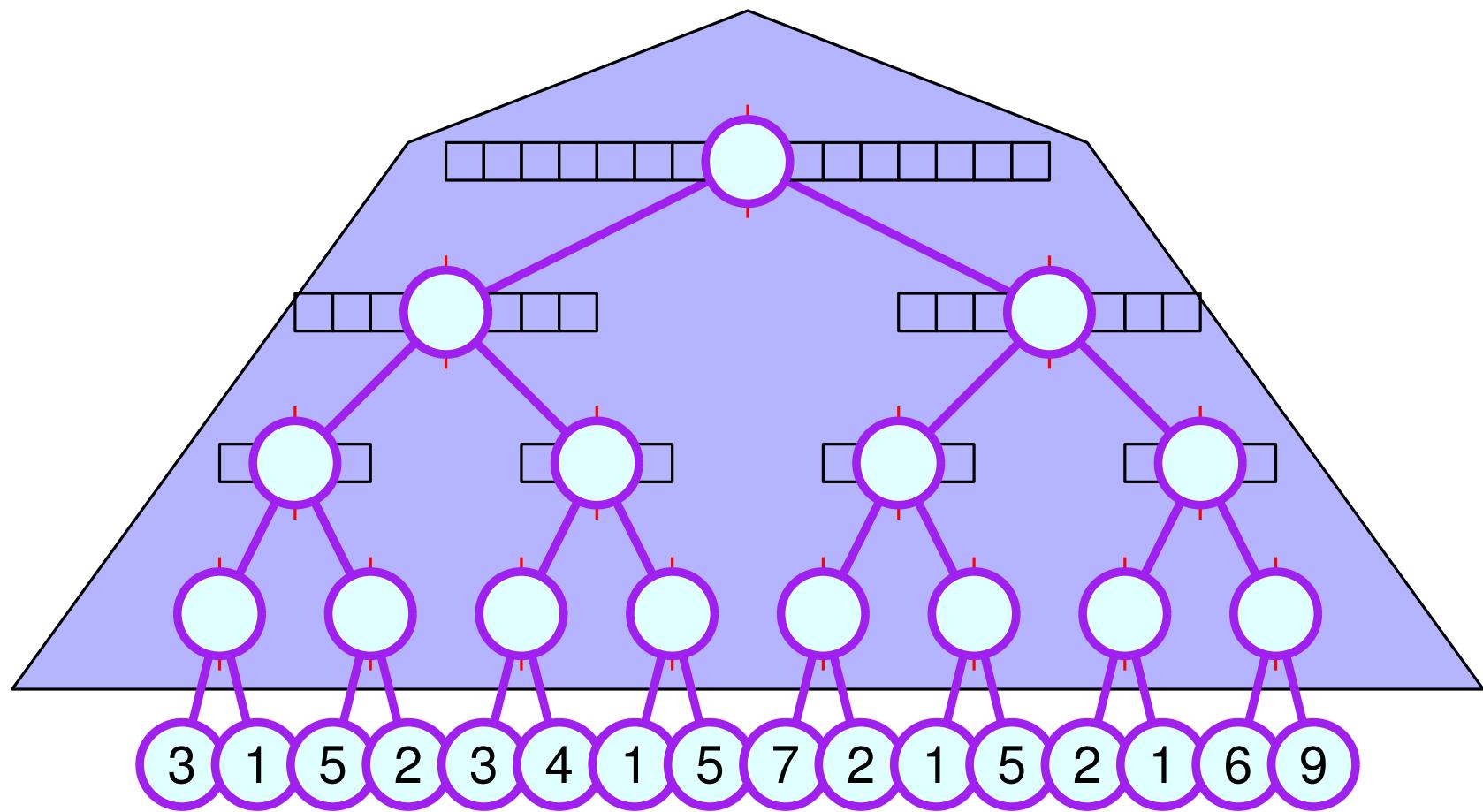
Work-efficient Prefix Sums



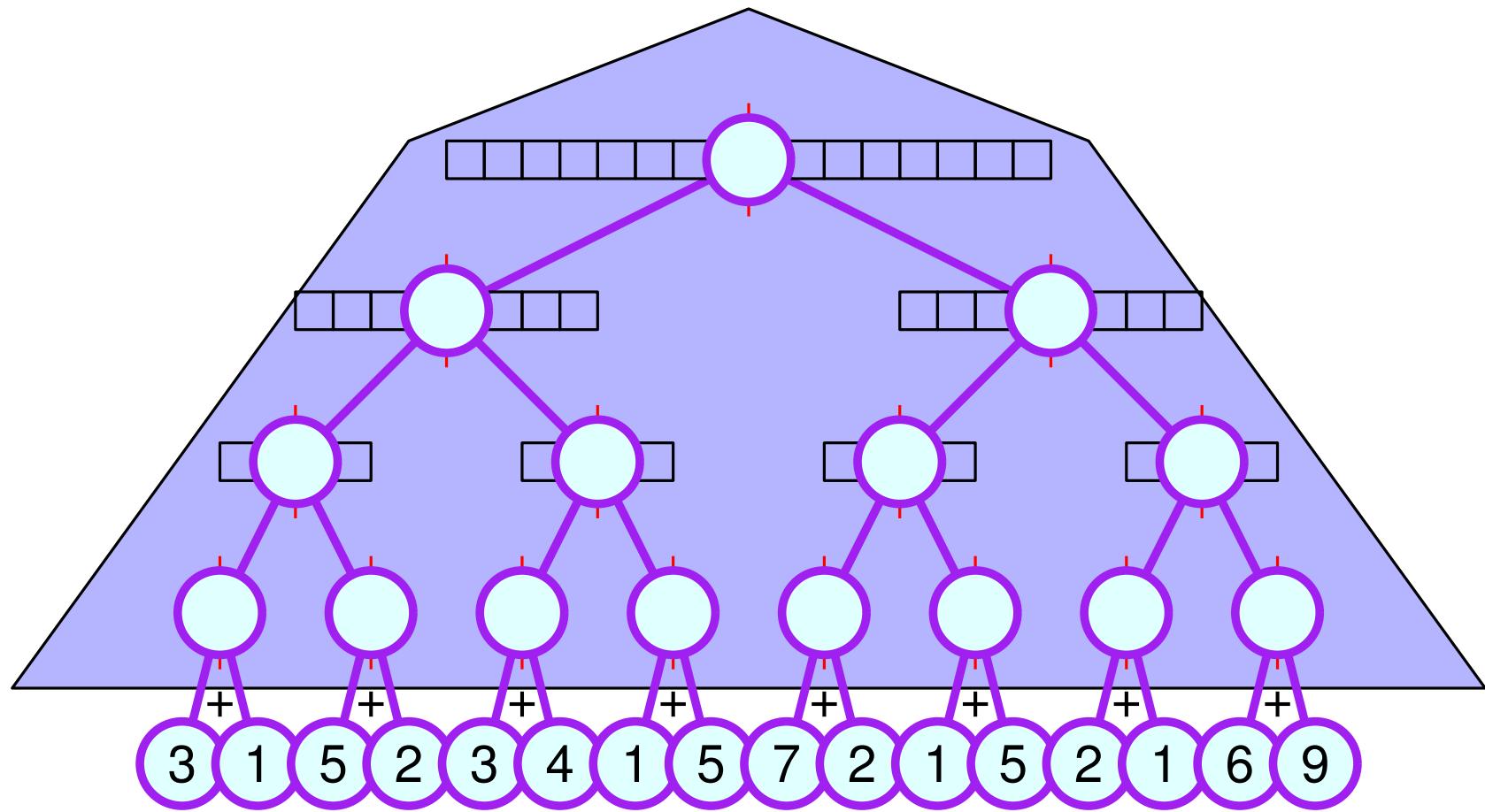
Work-efficient Prefix Sums



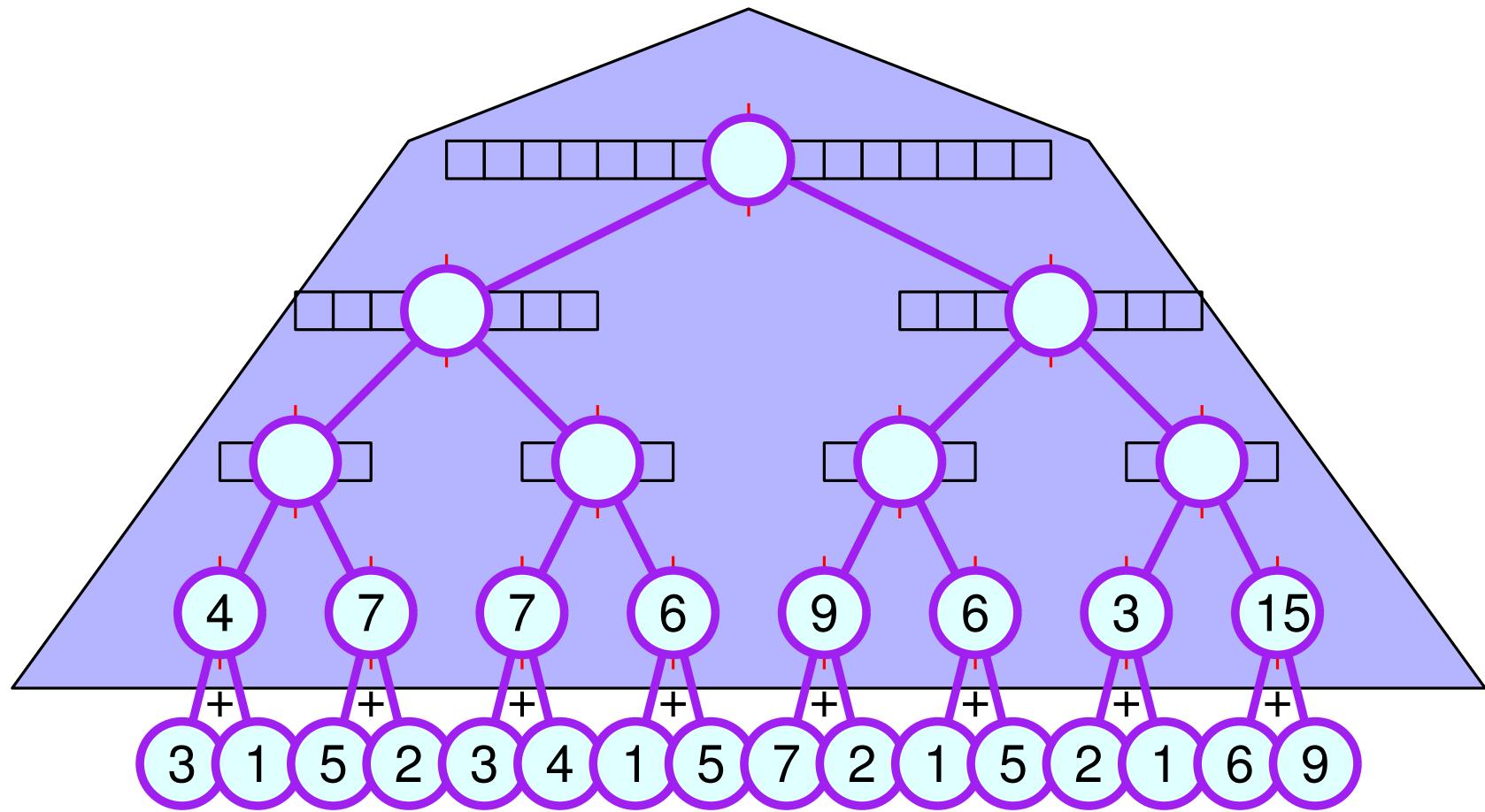
Work-efficient Prefix Sums



Work-efficient Prefix Sums



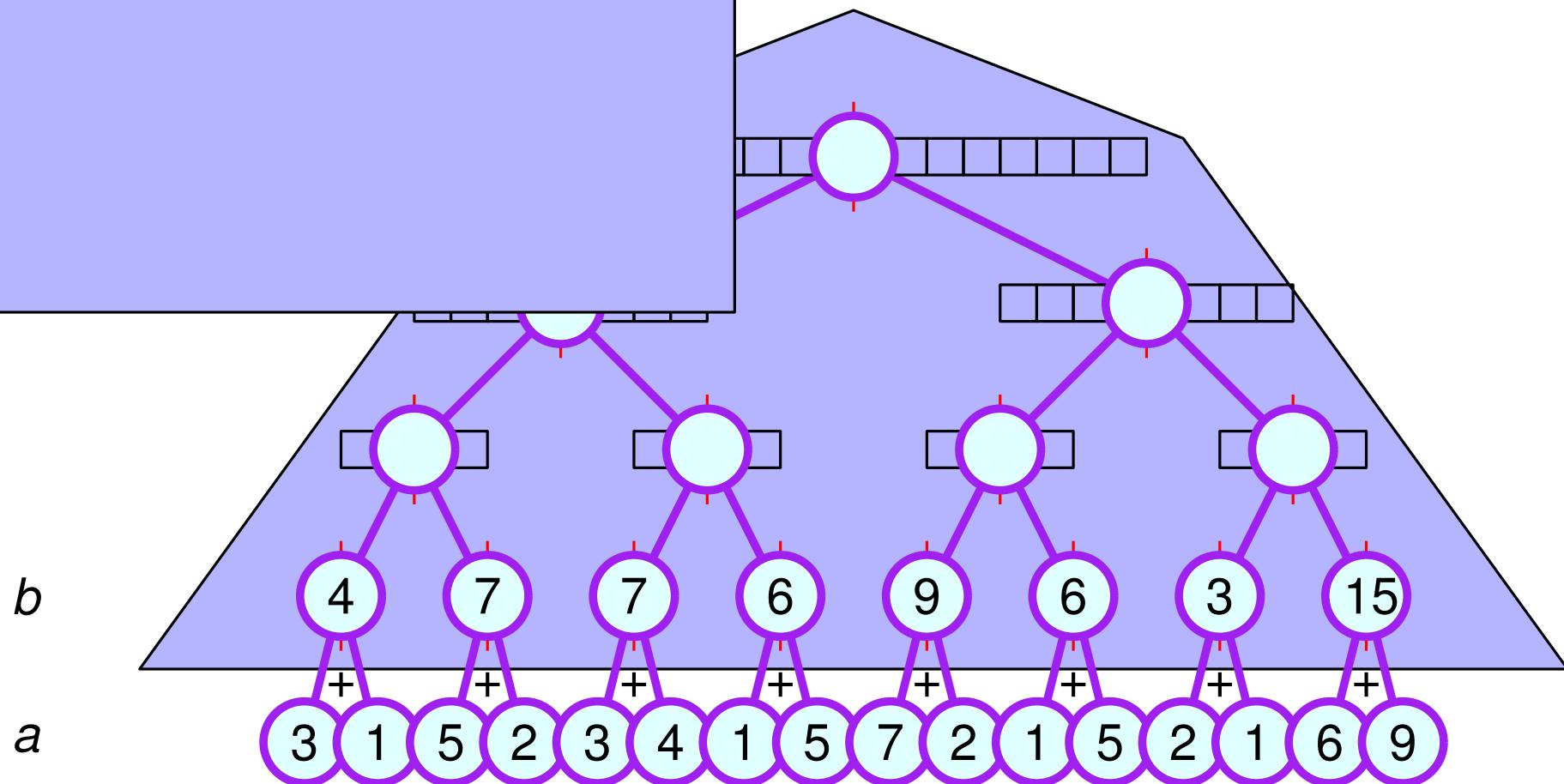
Work-efficient Prefix Sums



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
for  $i = 1$  to  $\frac{n}{2}$  in parallel do  
 $b[i] = a[2i - 1] + a[2i]$ 
```



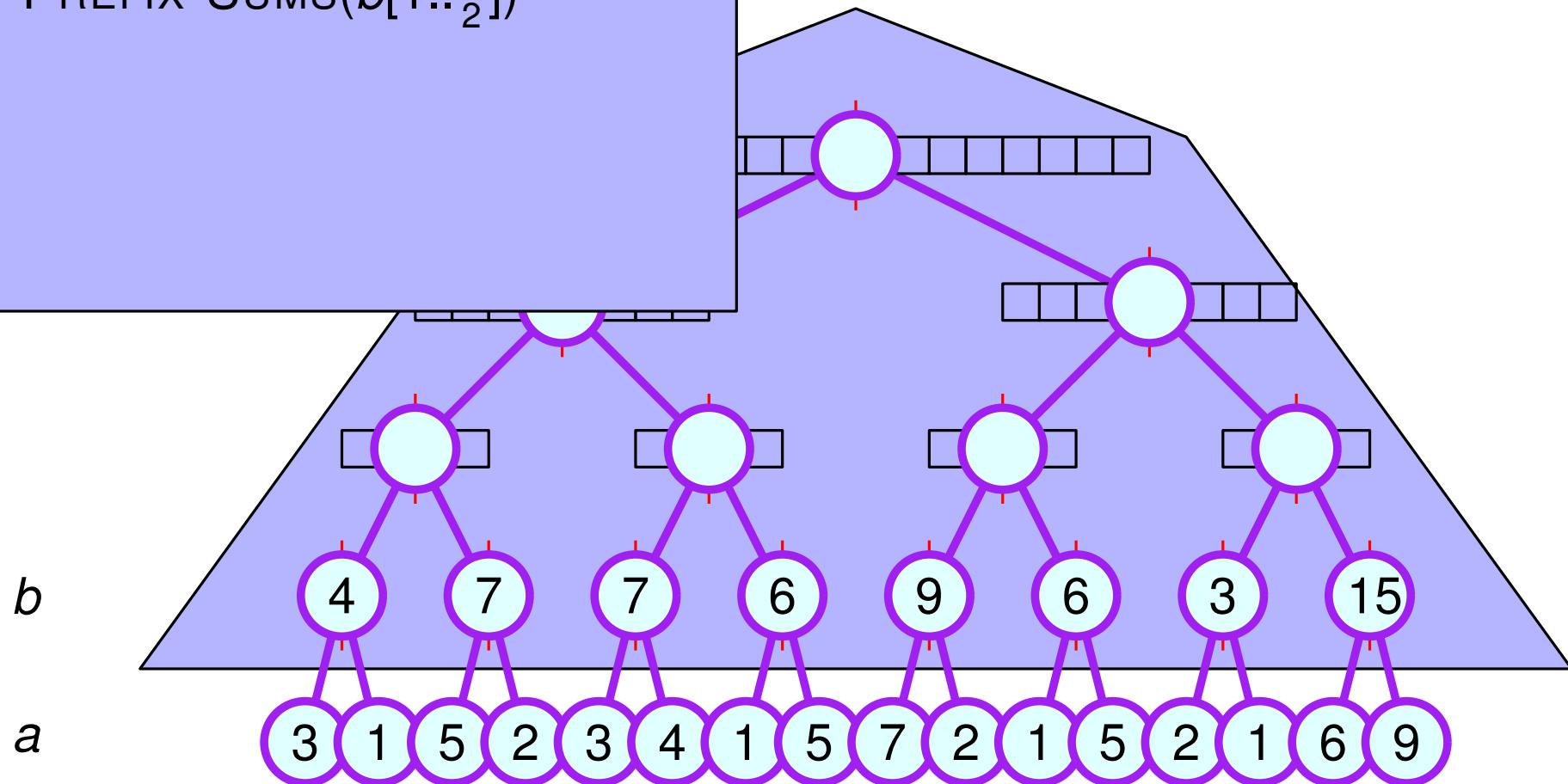
Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
     $b[i] = a[2i - 1] + a[2i]$ 
```

```
PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```



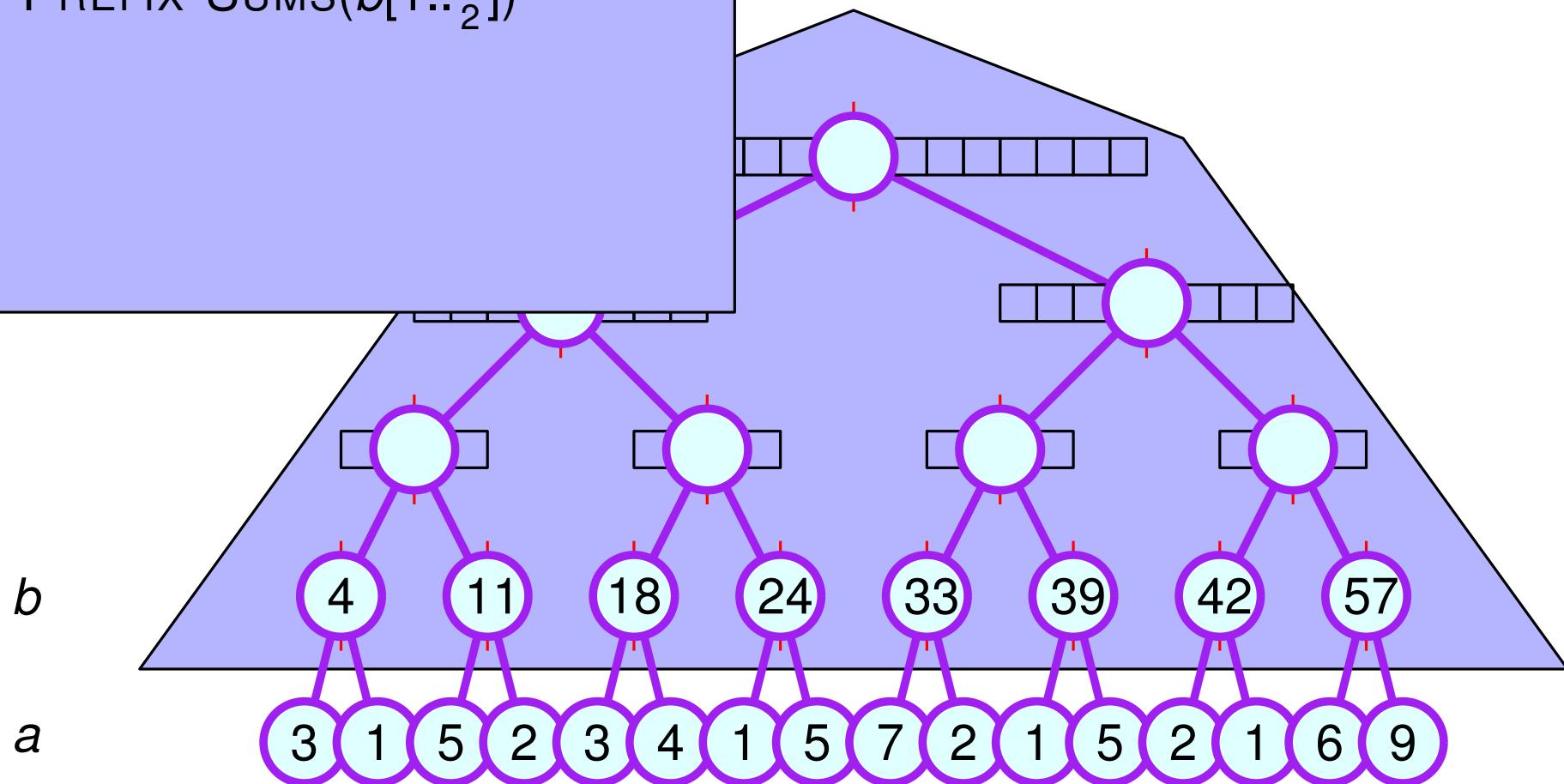
Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```



Work-efficient Prefix Sums

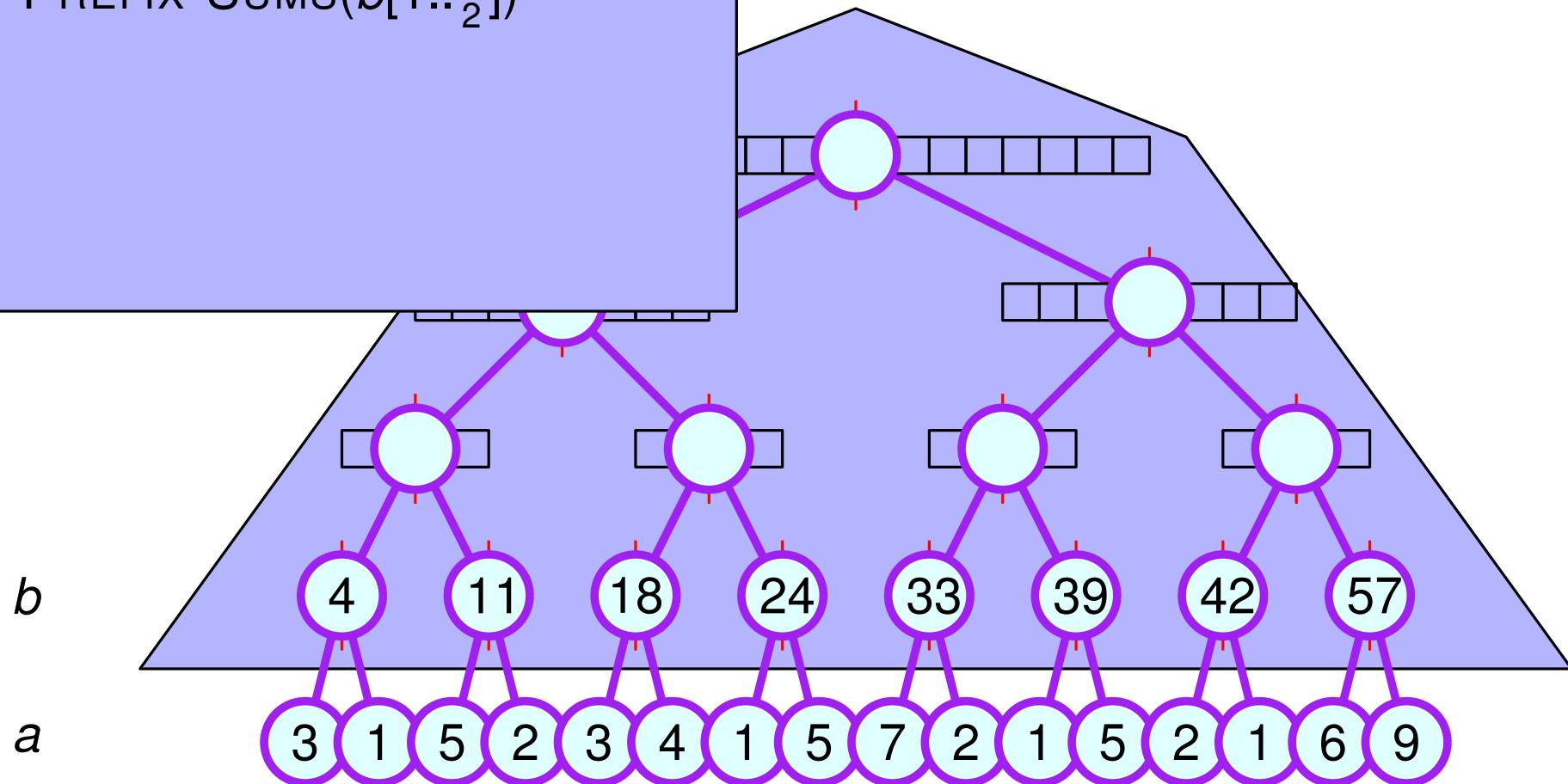
```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

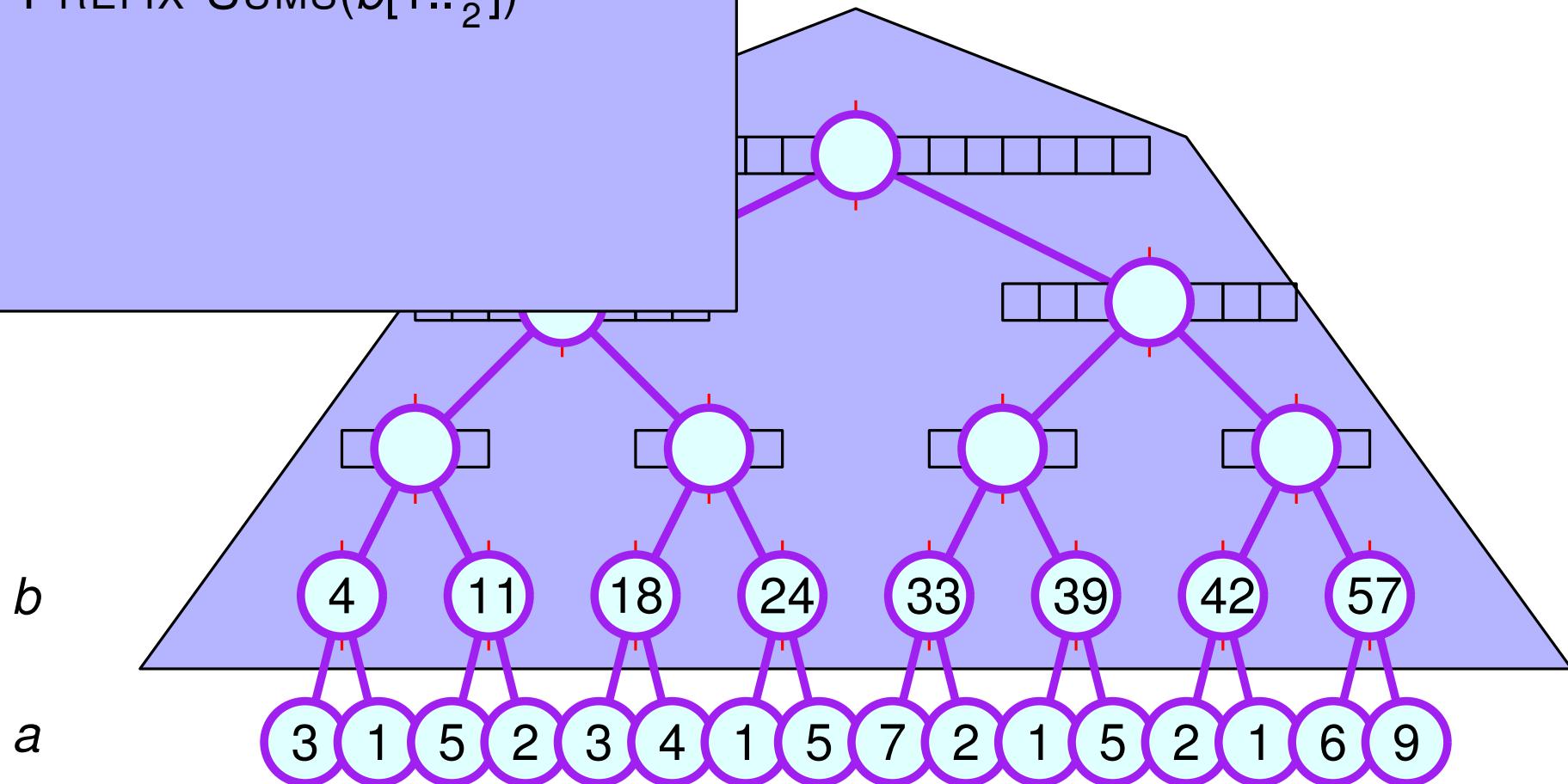
```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

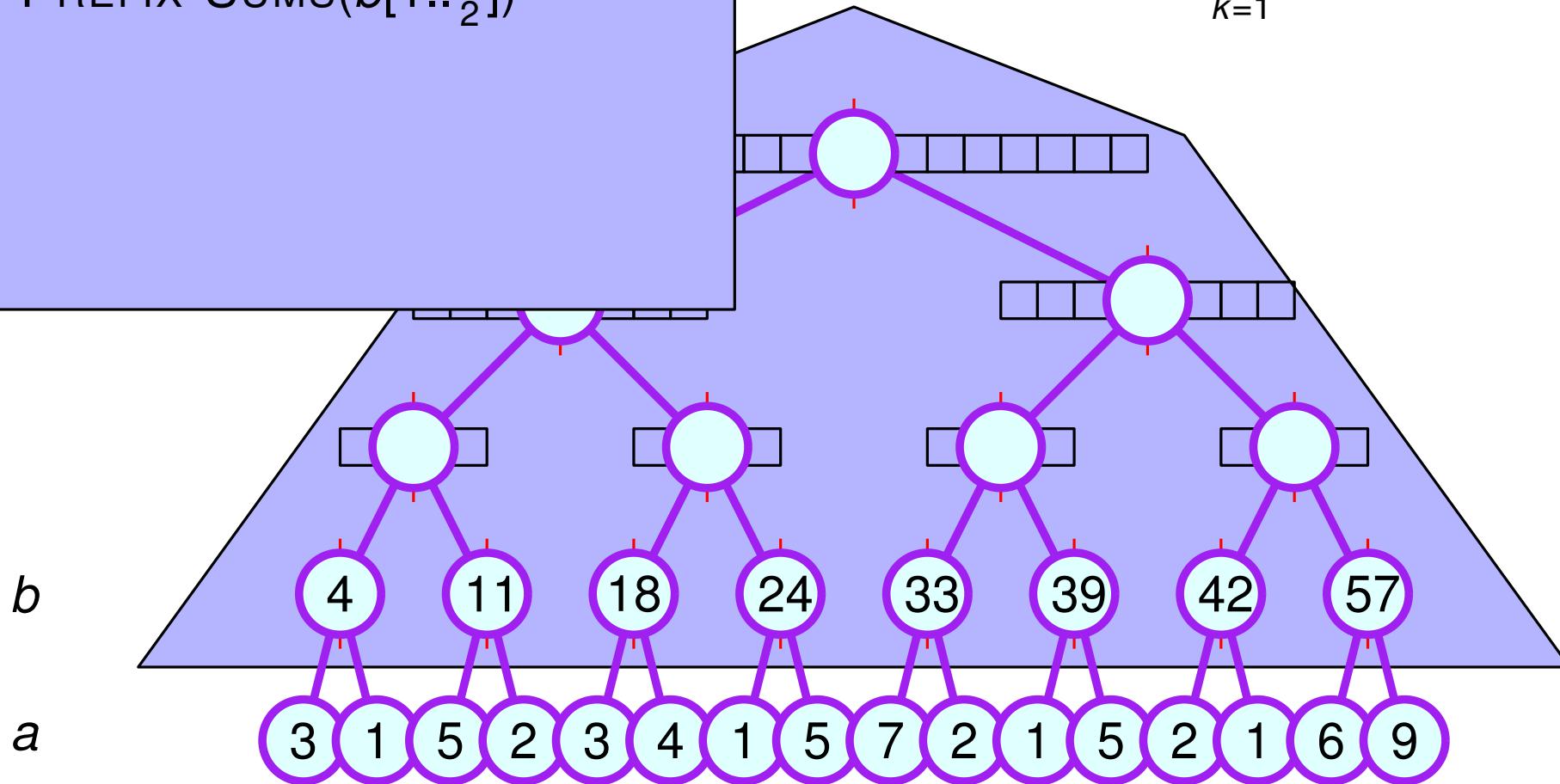
```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$
 $= \sum_{k=1}^i (a[2k - 1] + a[2k])$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

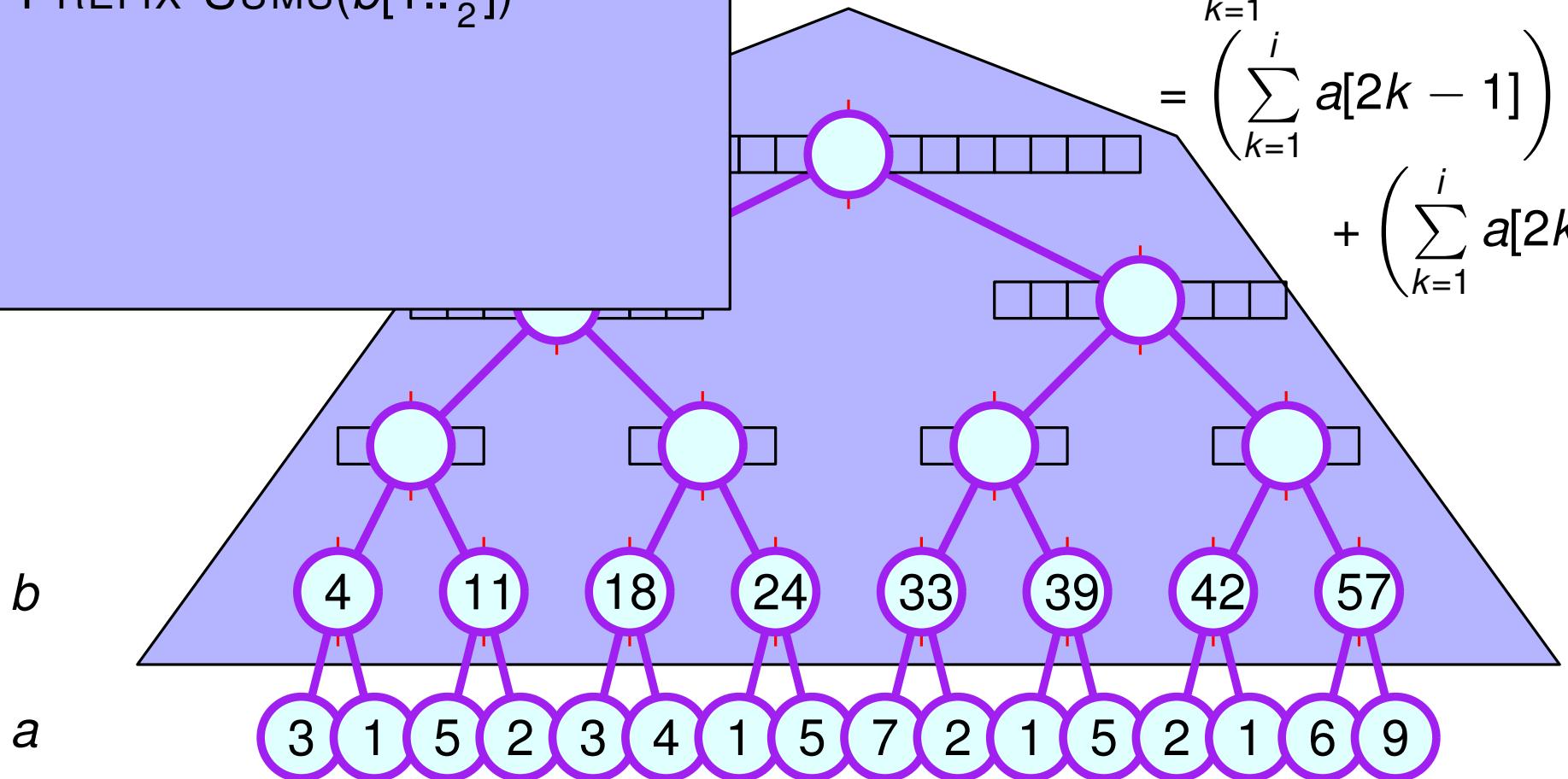
```
        PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right) + \left(\sum_{k=1}^i a[2k] \right)$$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
        PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

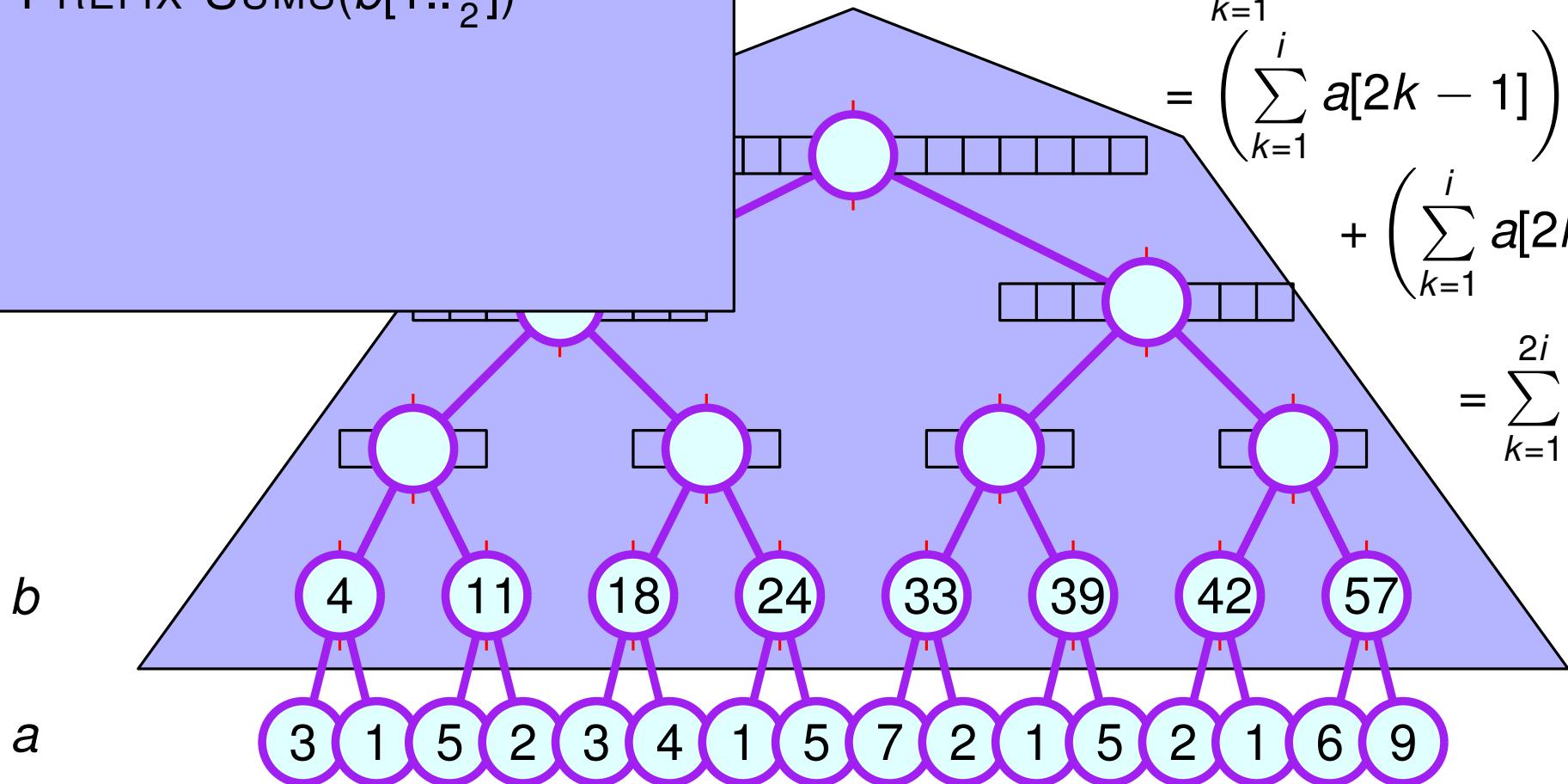
Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right)$$

$$+ \left(\sum_{k=1}^i a[2k] \right)$$

$$= \sum_{k=1}^{2i} a[k]$$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
        PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

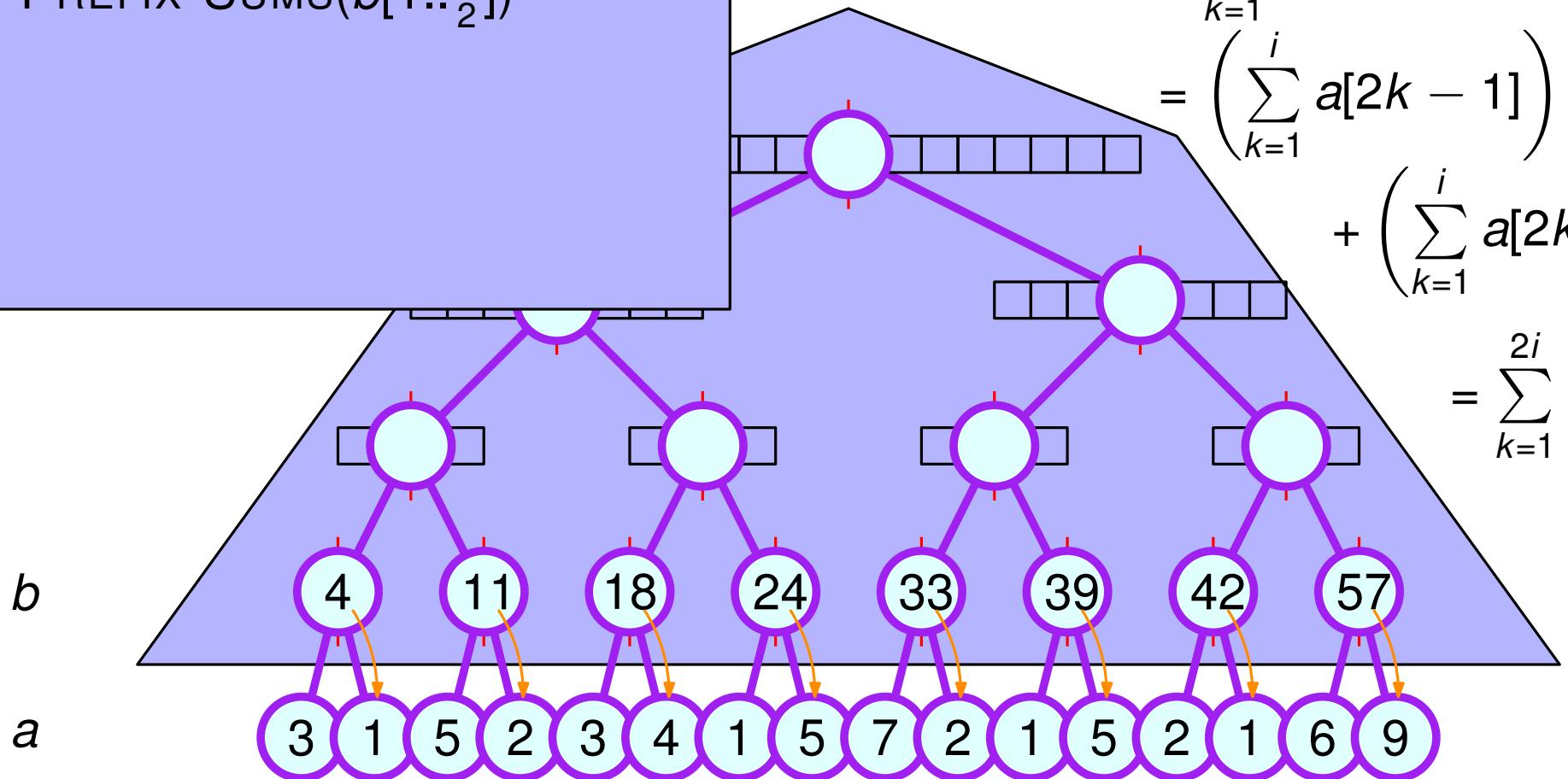
Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right)$$

$$+ \left(\sum_{k=1}^i a[2k] \right)$$

$$= \sum_{k=1}^{2i} a[k]$$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
        PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

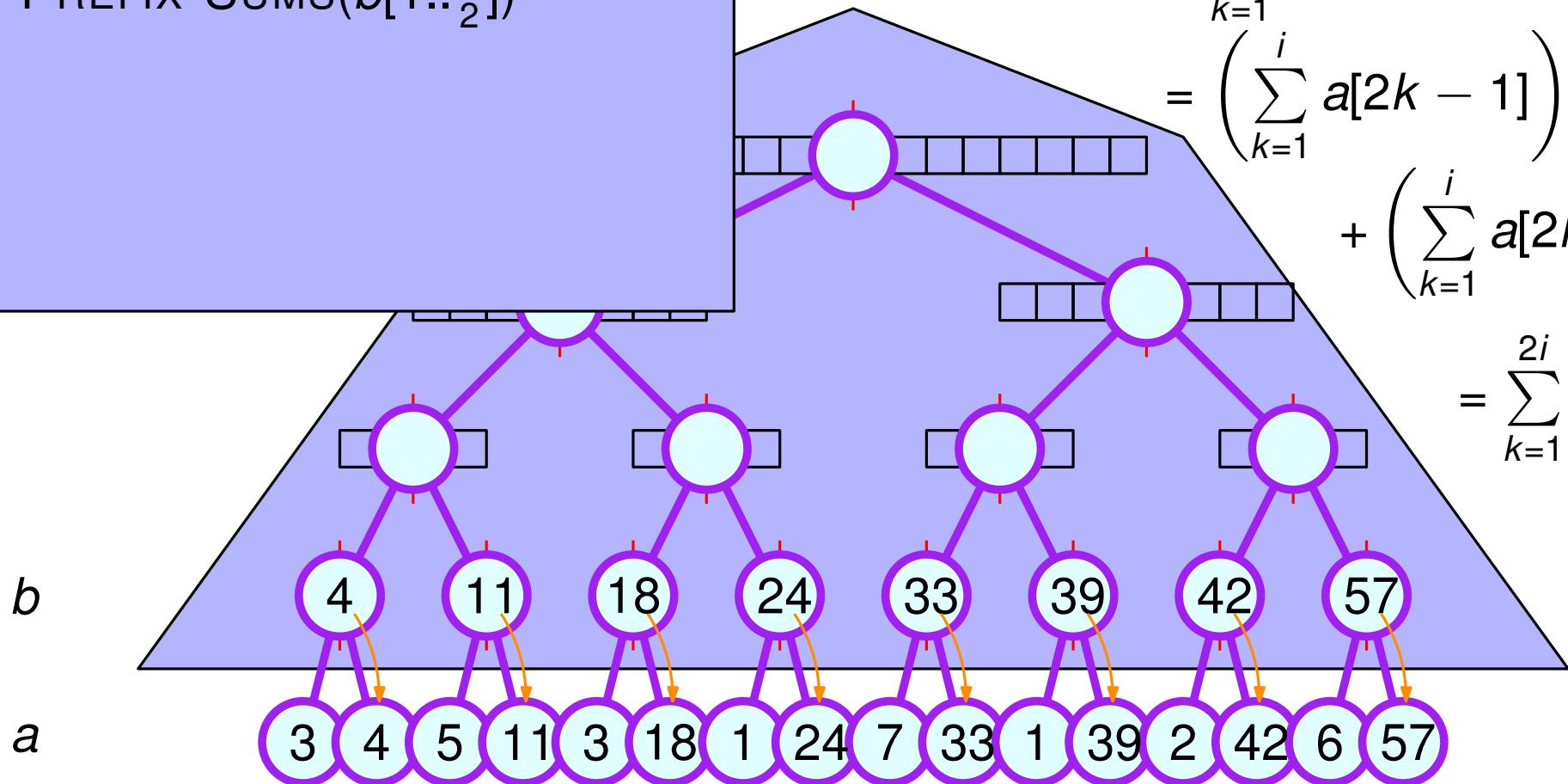
Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right)$$

$$+ \left(\sum_{k=1}^i a[2k] \right)$$

$$= \sum_{k=1}^{2i} a[k]$$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $a[2i] = b[i]$ 
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

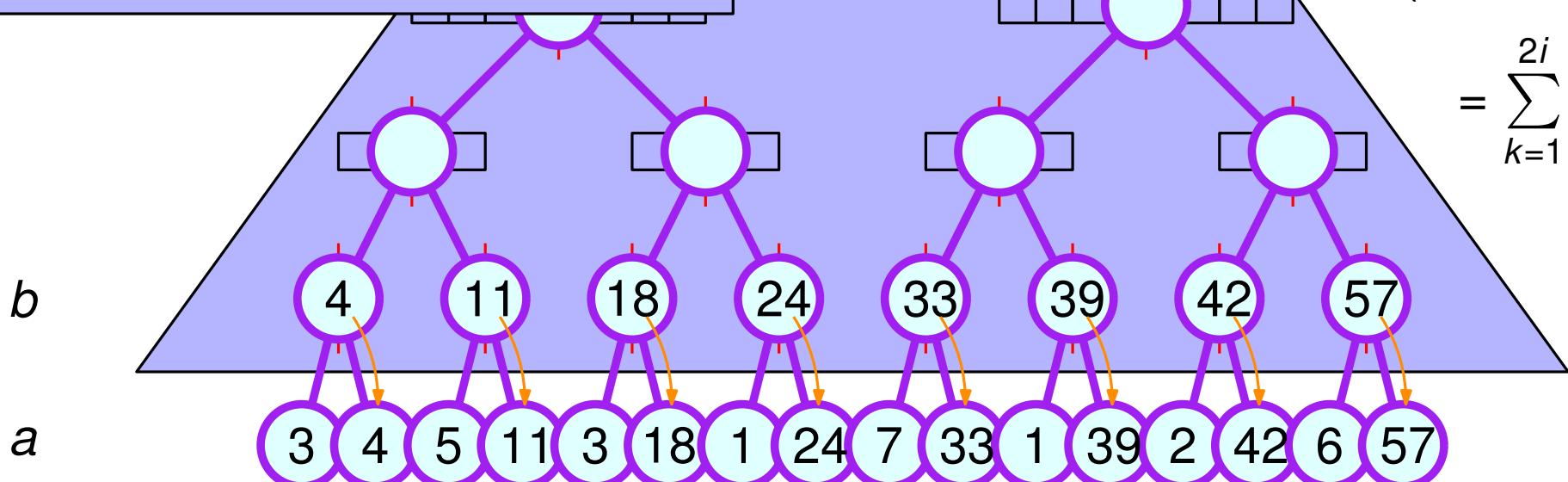
Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right)$$

$$+ \left(\sum_{k=1}^i a[2k] \right)$$

$$= \sum_{k=1}^{2i} a[k]$$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $a[2i] = b[i]$ 
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

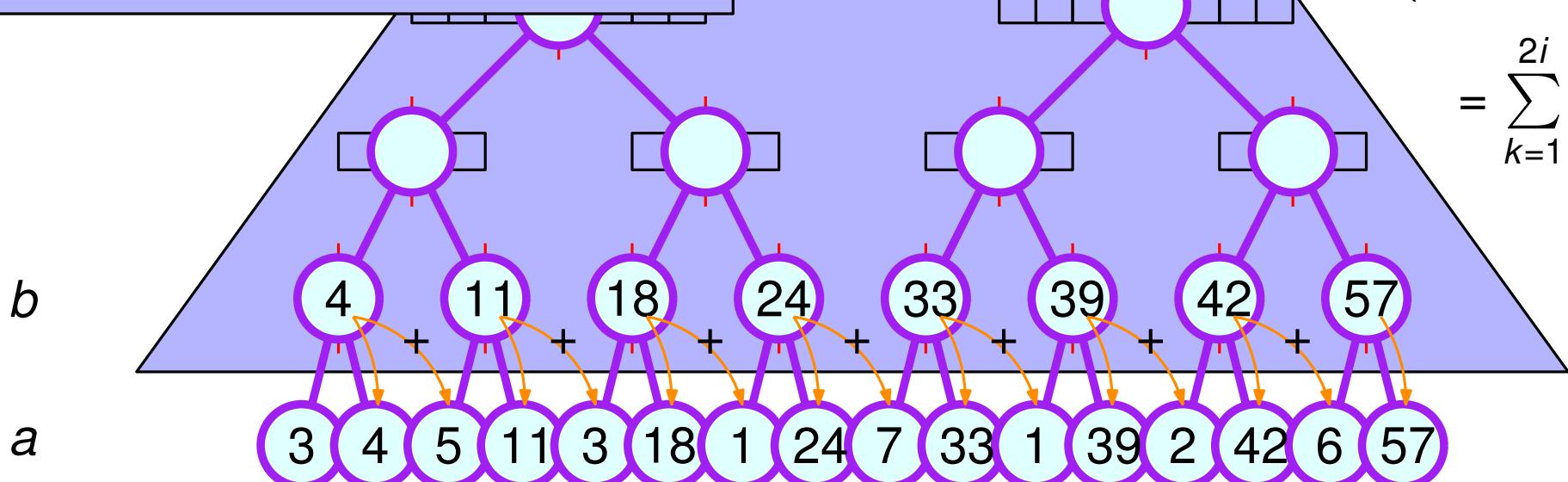
Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right)$$

$$+ \left(\sum_{k=1}^i a[2k] \right)$$

$$= \sum_{k=1}^{2i} a[k]$$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $a[2i] = b[i]$ 
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

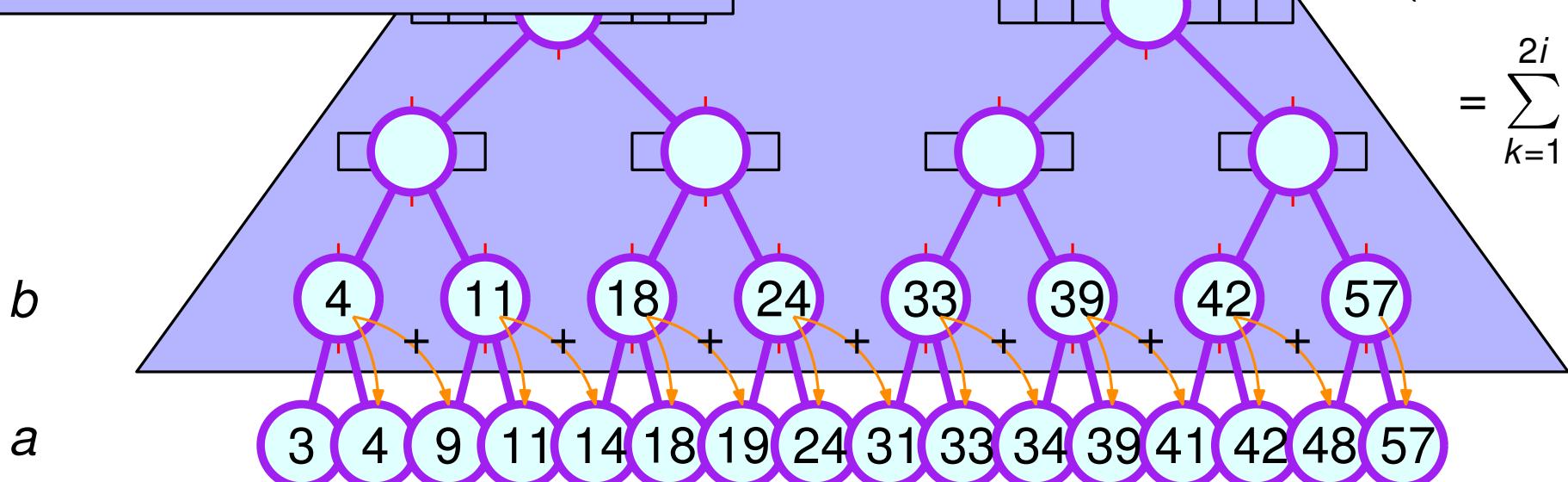
Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right)$$

$$+ \left(\sum_{k=1}^i a[2k] \right)$$

$$= \sum_{k=1}^{2i} a[k]$$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $a[2i] = b[i]$ 
```

```
         $a[2i + 1] = a[2i + 1] + b[i]$ 
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

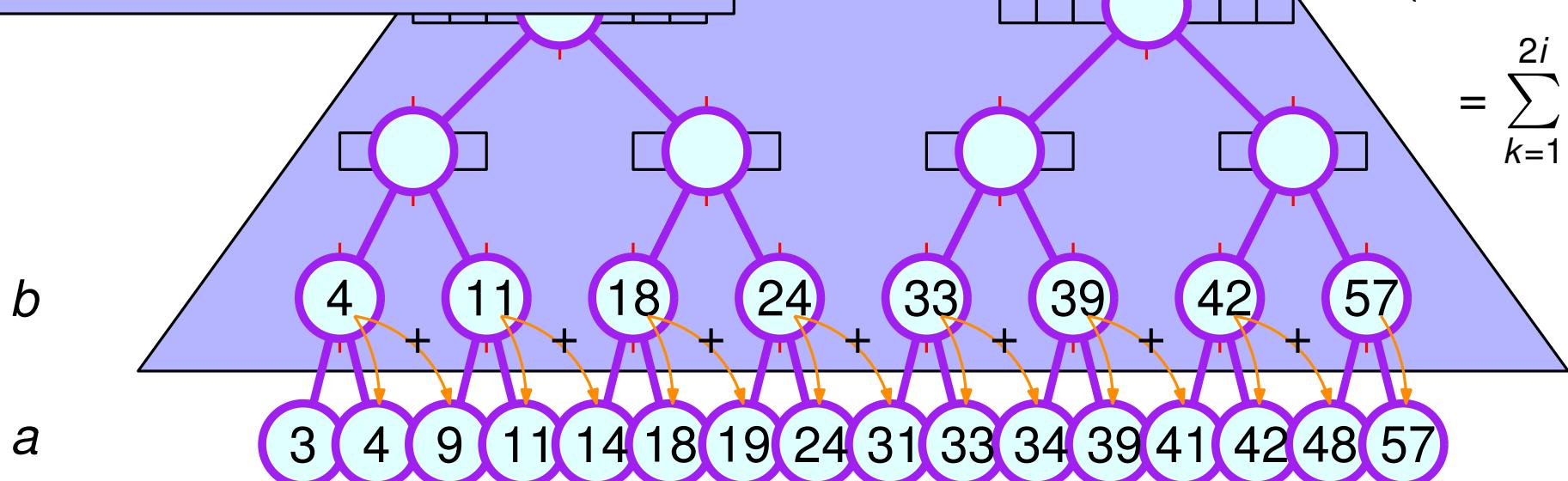
Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right)$$

$$+ \left(\sum_{k=1}^i a[2k] \right)$$

$$= \sum_{k=1}^{2i} a[k]$$



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $b[i] = a[2i - 1] + a[2i]$ 
```

```
PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
```

```
    for  $i = 1$  to  $\frac{n}{2}$  in parallel do
```

```
         $a[2i] = b[i]$ 
```

```
        if  $i \neq \frac{n}{2}$  then
```

```
             $a[2i + 1] = a[2i + 1] + b[i]$ 
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

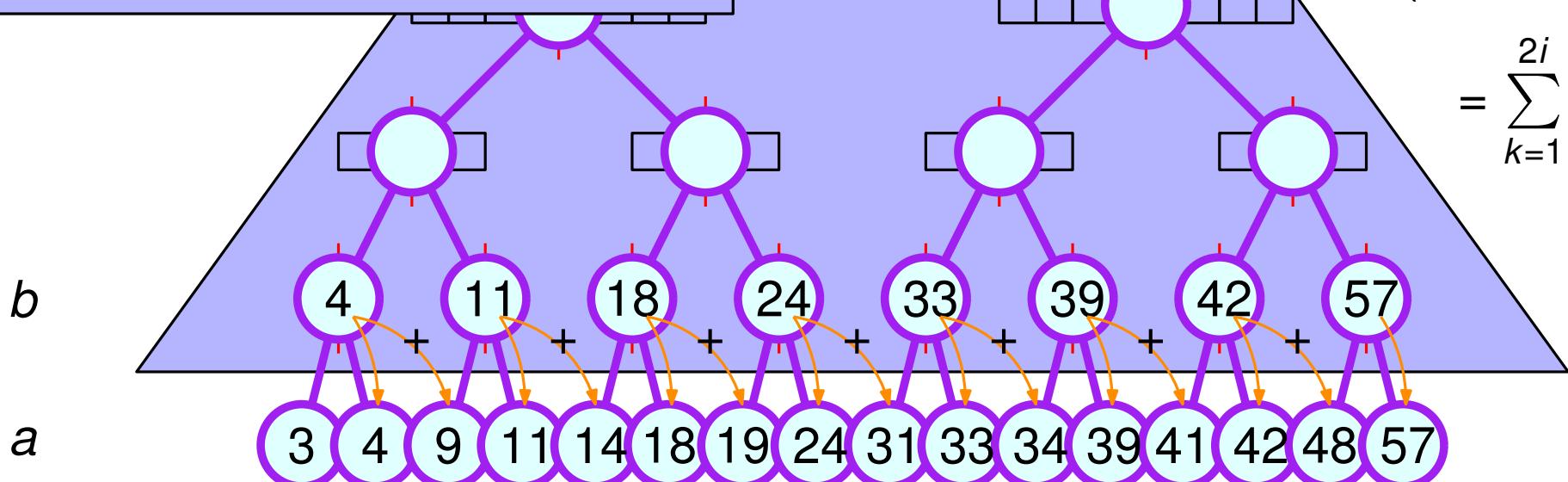
Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$= \sum_{k=1}^i (a[2k - 1] + a[2k])$$

$$= \left(\sum_{k=1}^i a[2k - 1] \right)$$

$$+ \left(\sum_{k=1}^i a[2k] \right)$$

$$= \sum_{k=1}^{2i} a[k]$$



Work-efficient Prefix Sums

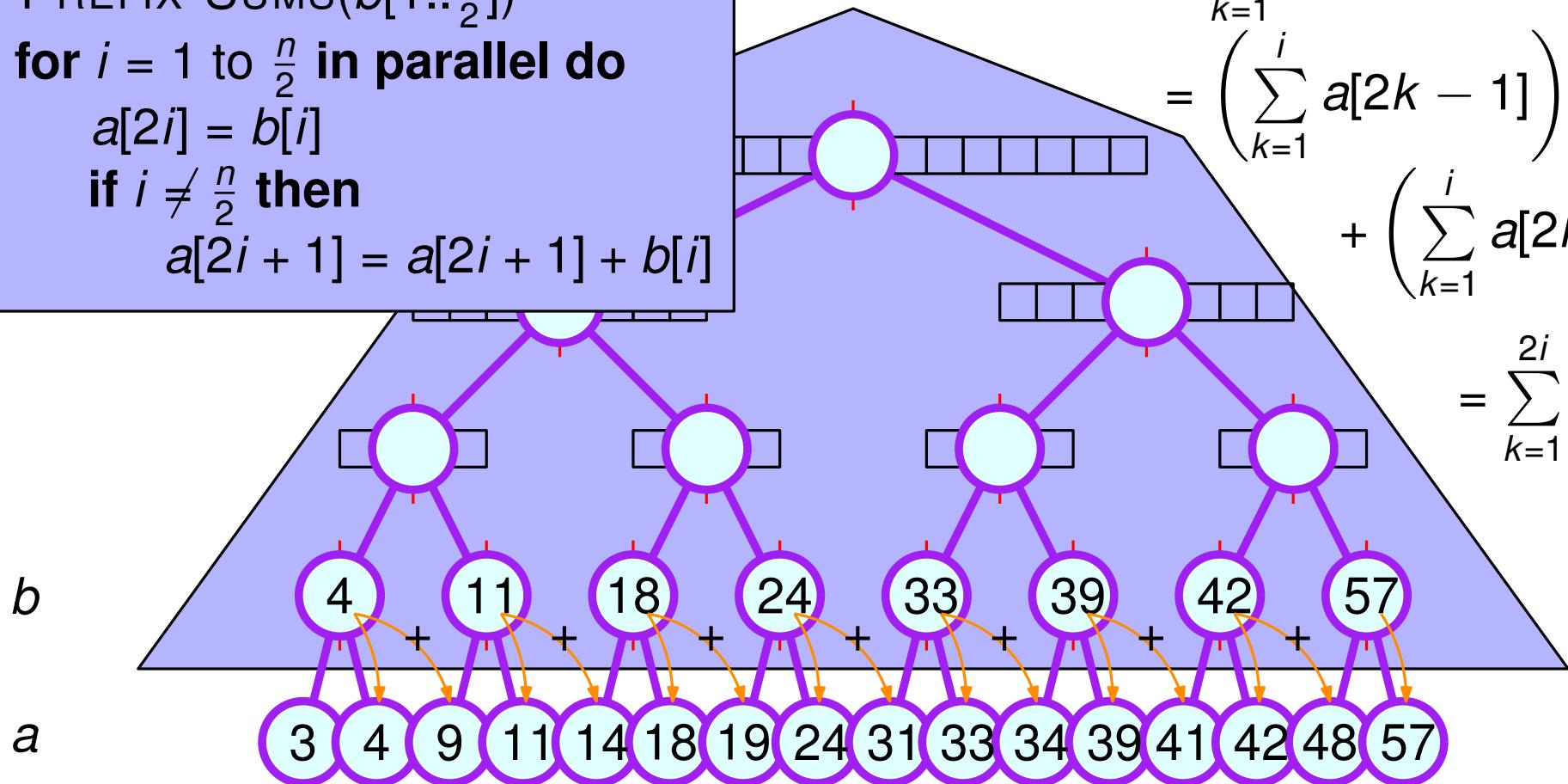
```

procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
  
```

Claim. $b[i] = \sum_{k=1}^{2i} a[k]$

Proof. By I.H. $b[i] = \sum_{k=1}^i b[k]$

$$\begin{aligned}
 &= \sum_{k=1}^i (a[2k - 1] + a[2k]) \\
 &= \left(\sum_{k=1}^i a[2k - 1] \right) \\
 &\quad + \left(\sum_{k=1}^i a[2k] \right) \\
 &= \sum_{k=1}^{2i} a[k]
 \end{aligned}$$



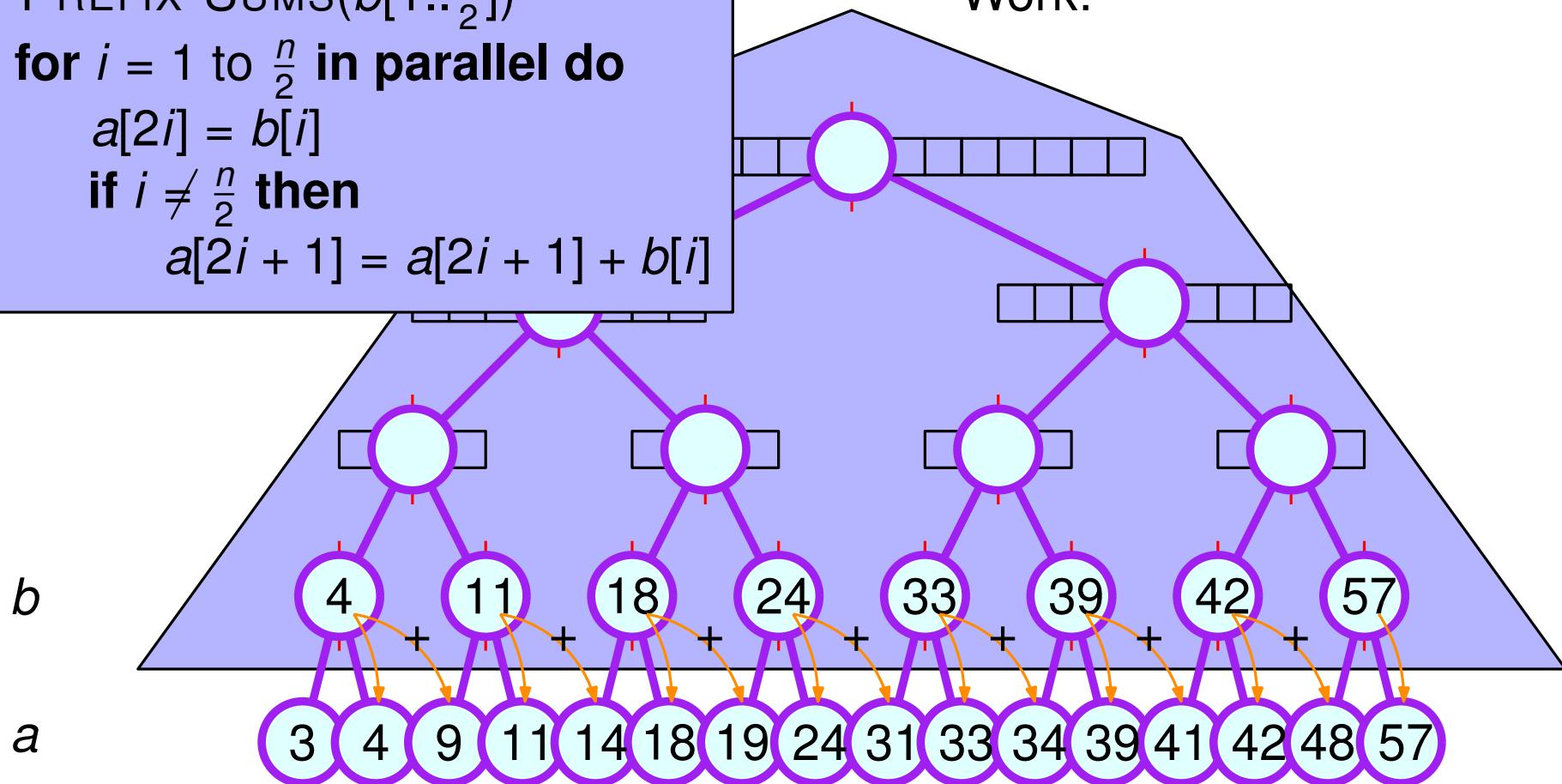
Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

Analysis

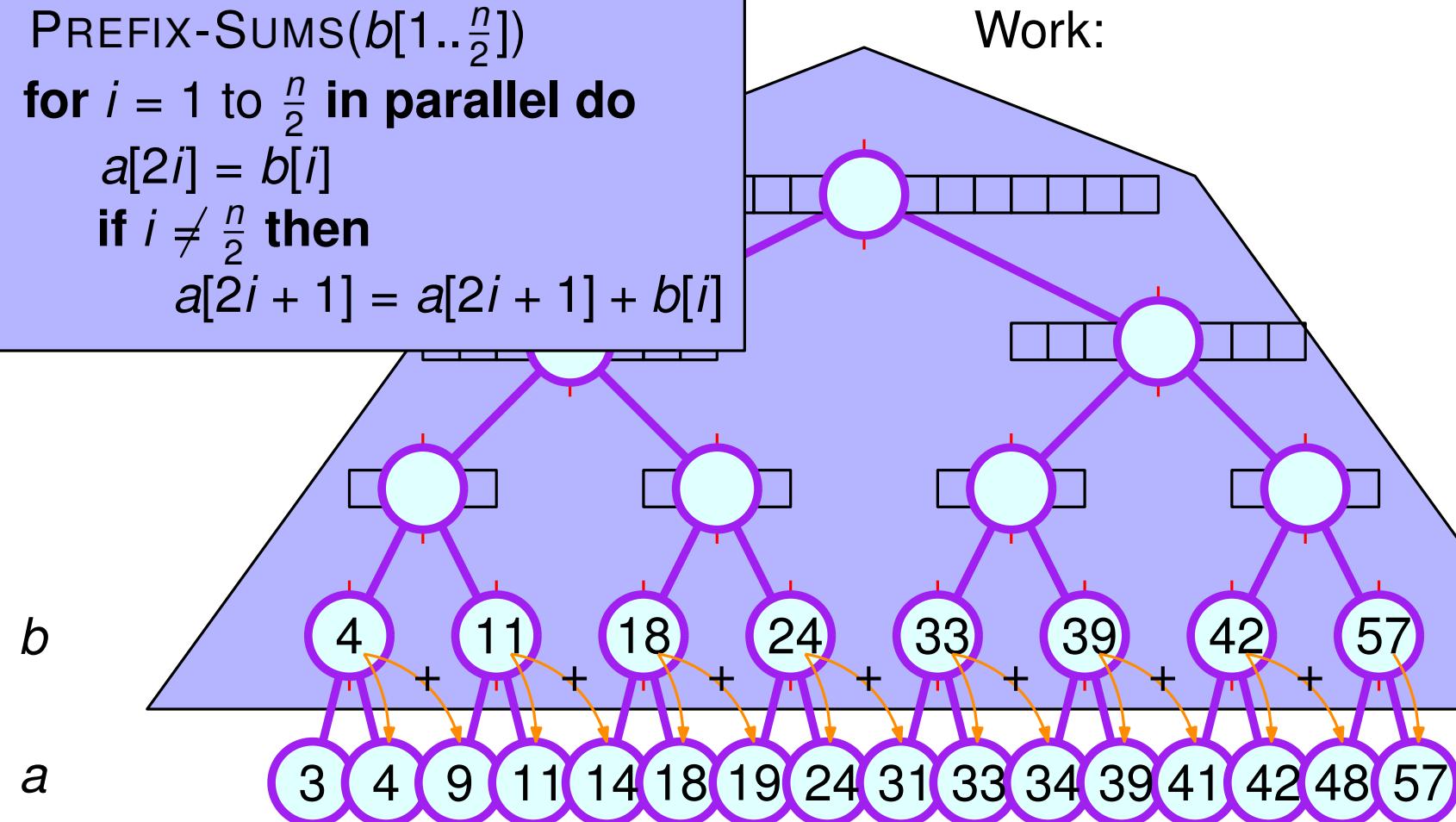
Time:

Work:



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```



Analysis

Time:

$$T(n) = T(n/2) + O(1)$$

Work:

Work-efficient Prefix Sums

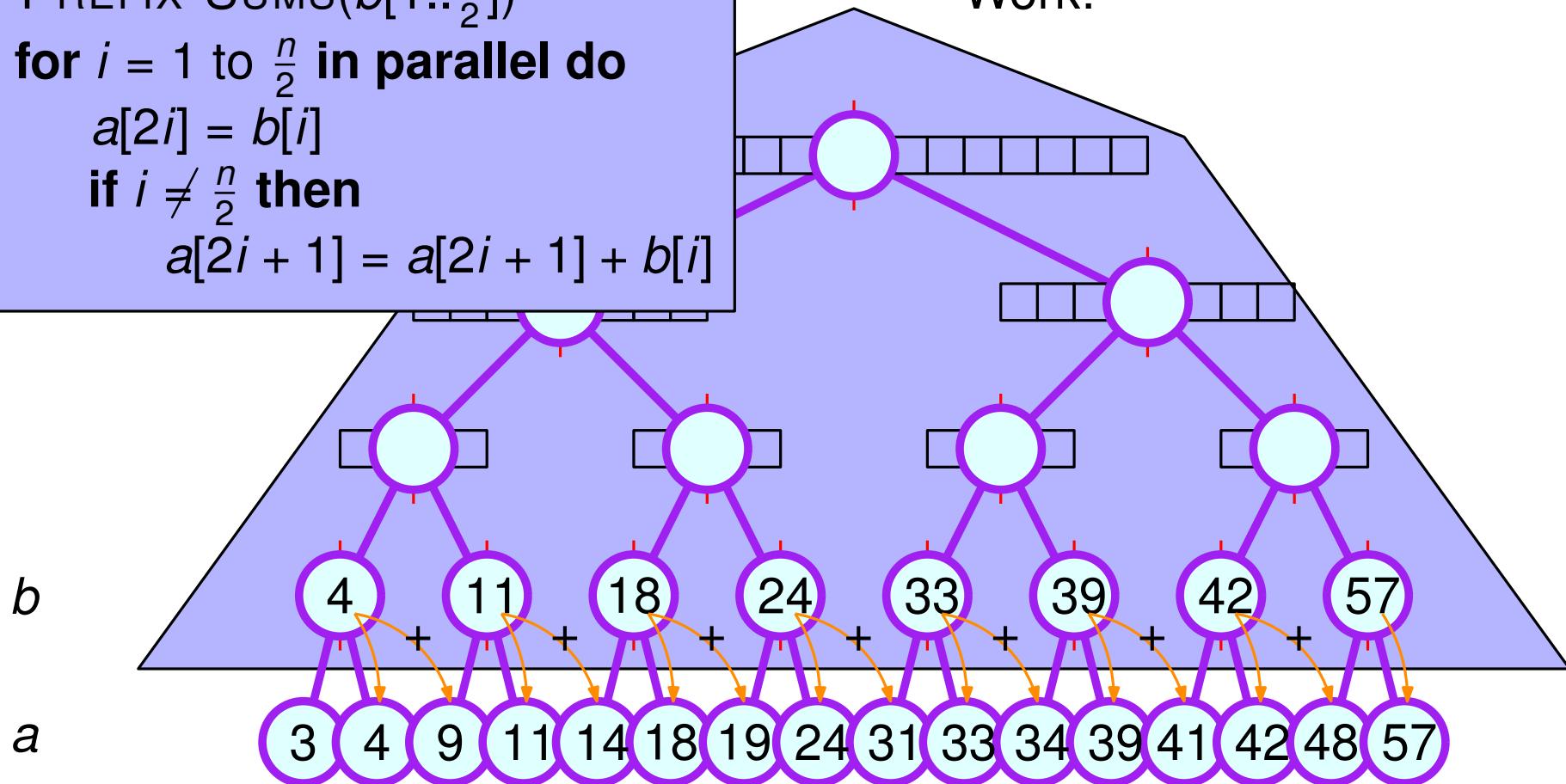
```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

Analysis

Time:

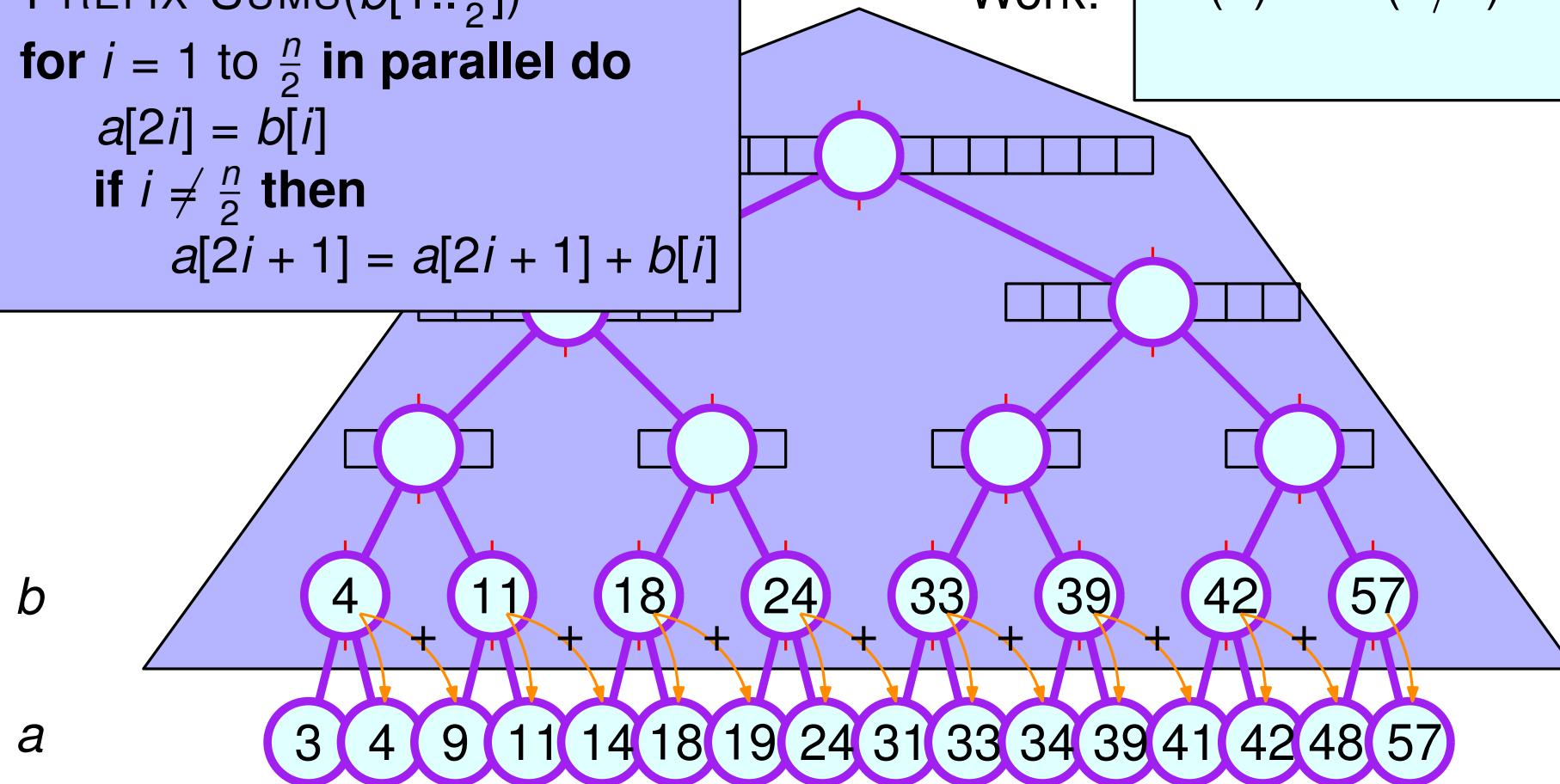
$$T(n) = T(n/2) + O(1)$$
$$= O(\log n)$$

Work:



Work-efficient Prefix Sums

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```



Analysis

Time:

$$T(n) = T(n/2) + O(1) \\ = O(\log n)$$

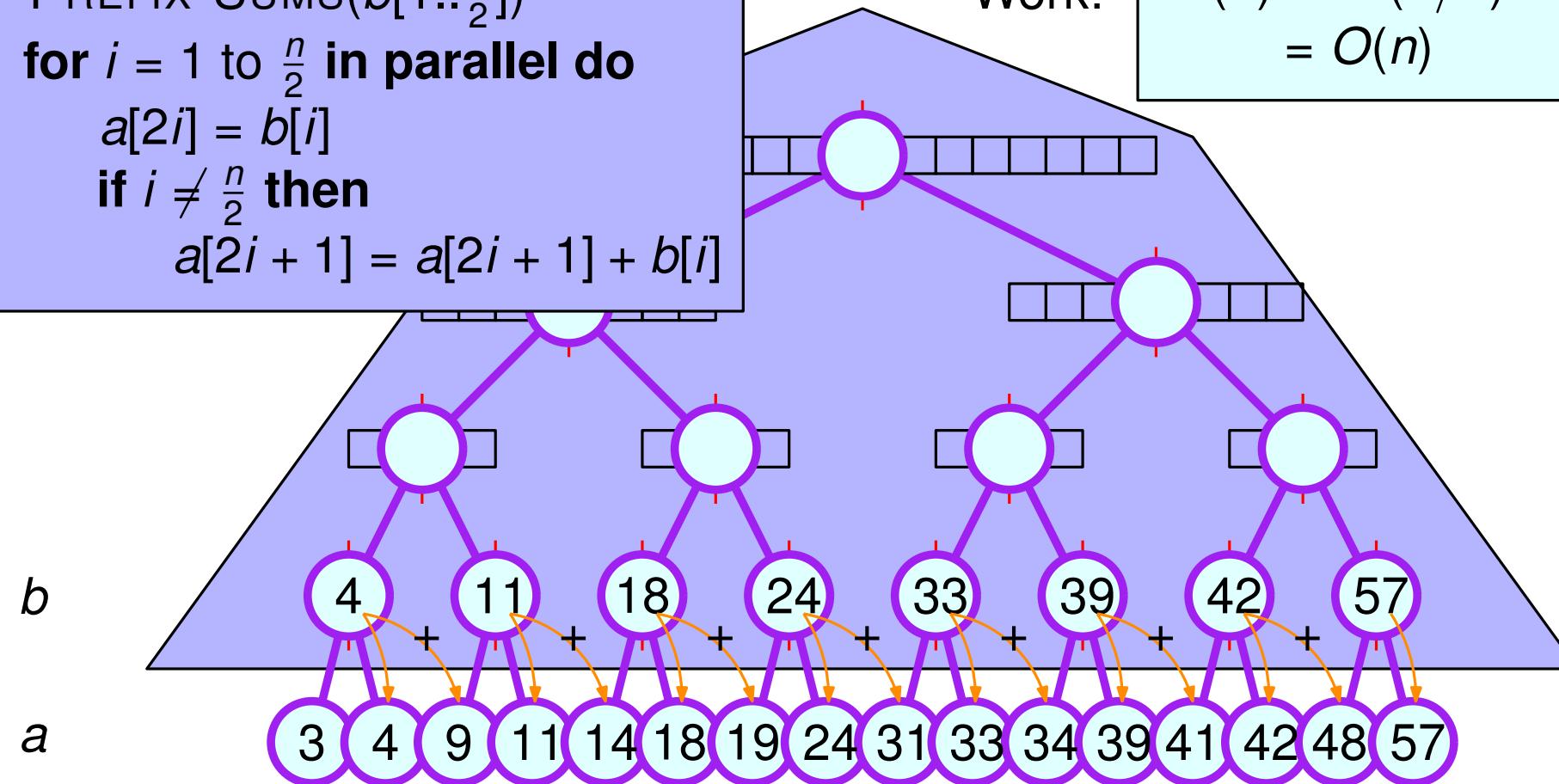
Work:

$$W(n) = W(n/2) + O(n)$$

Work-efficient Prefix Sums

```

procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
  
```



Analysis

Time:

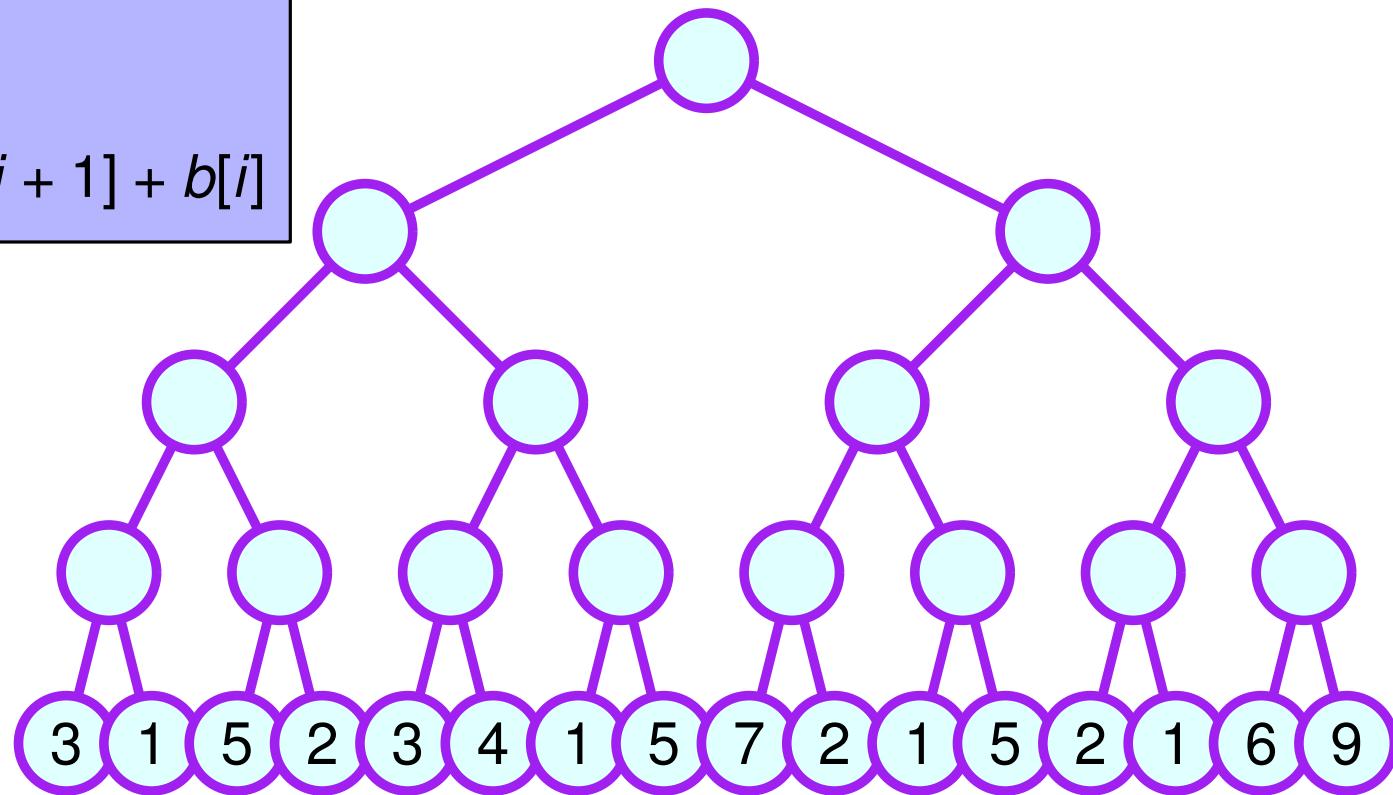
$$T(n) = T(n/2) + O(1) \\ = O(\log n)$$

Work:

$$W(n) = W(n/2) + O(n) \\ = O(n)$$

Balanced-Tree Technique

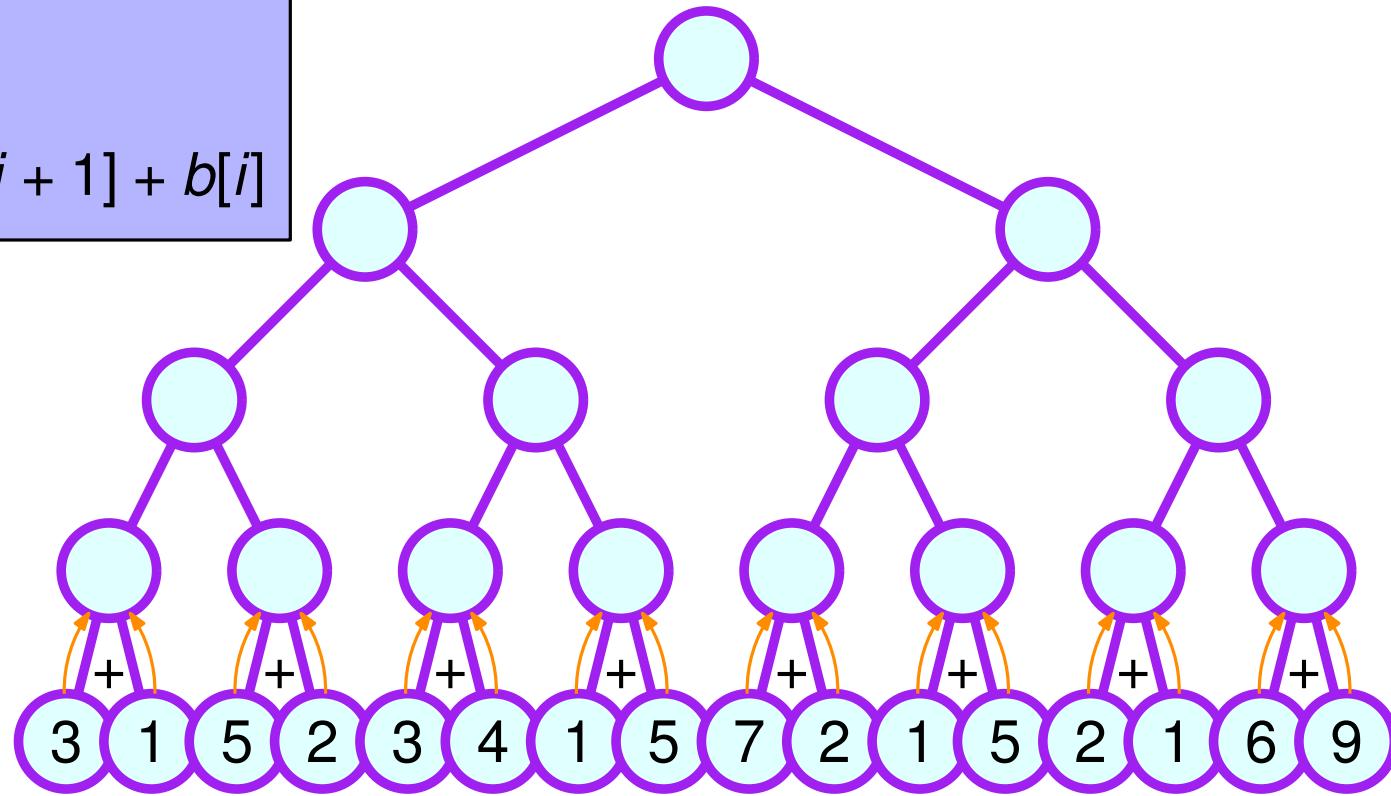
```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

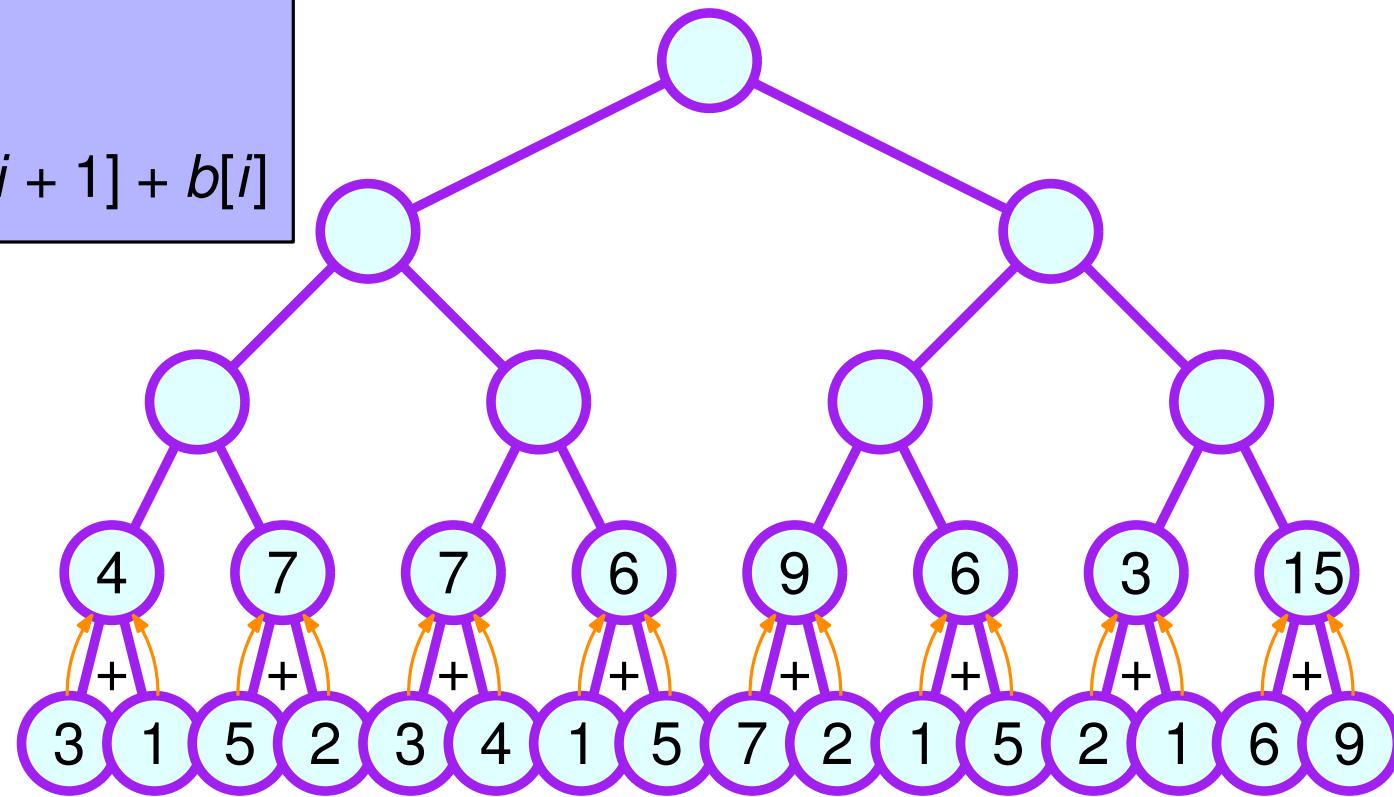
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

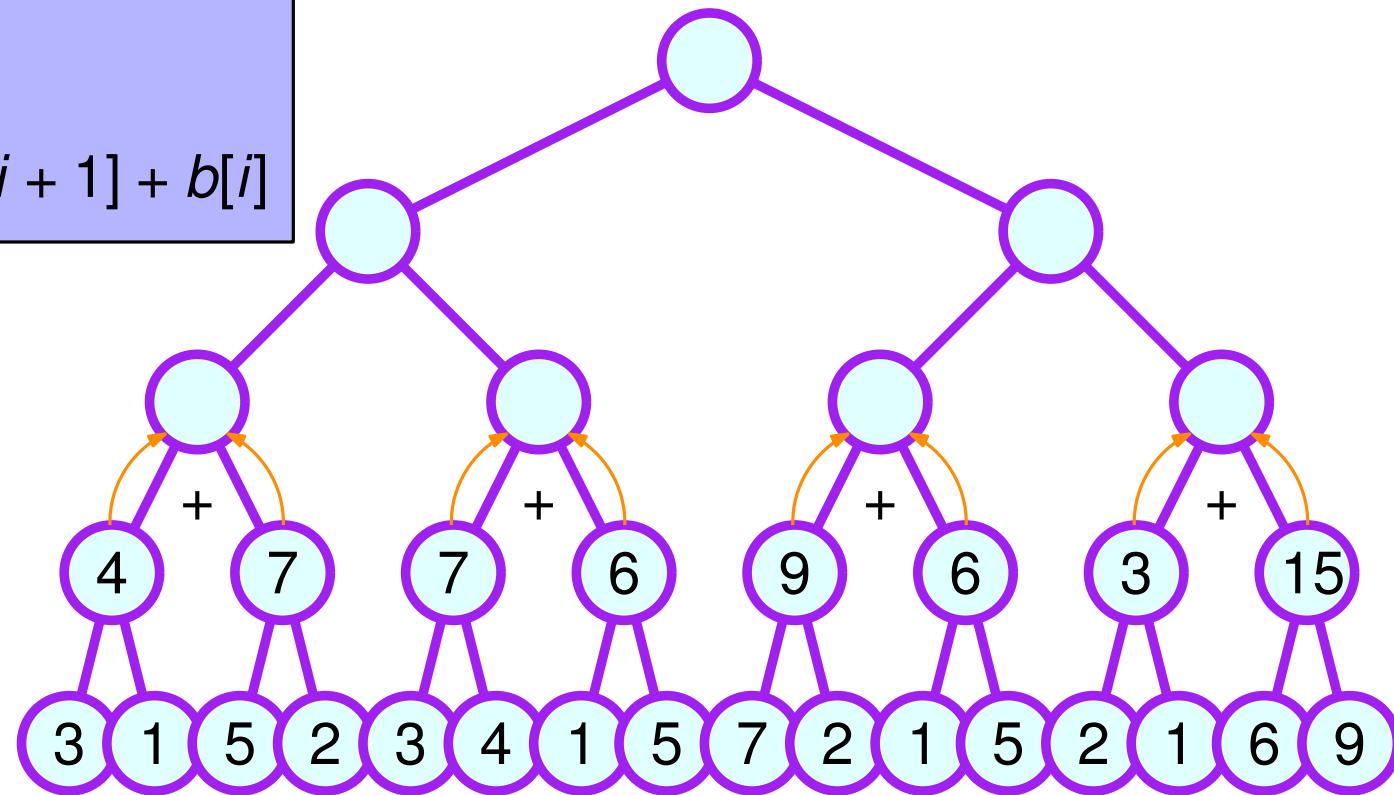
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

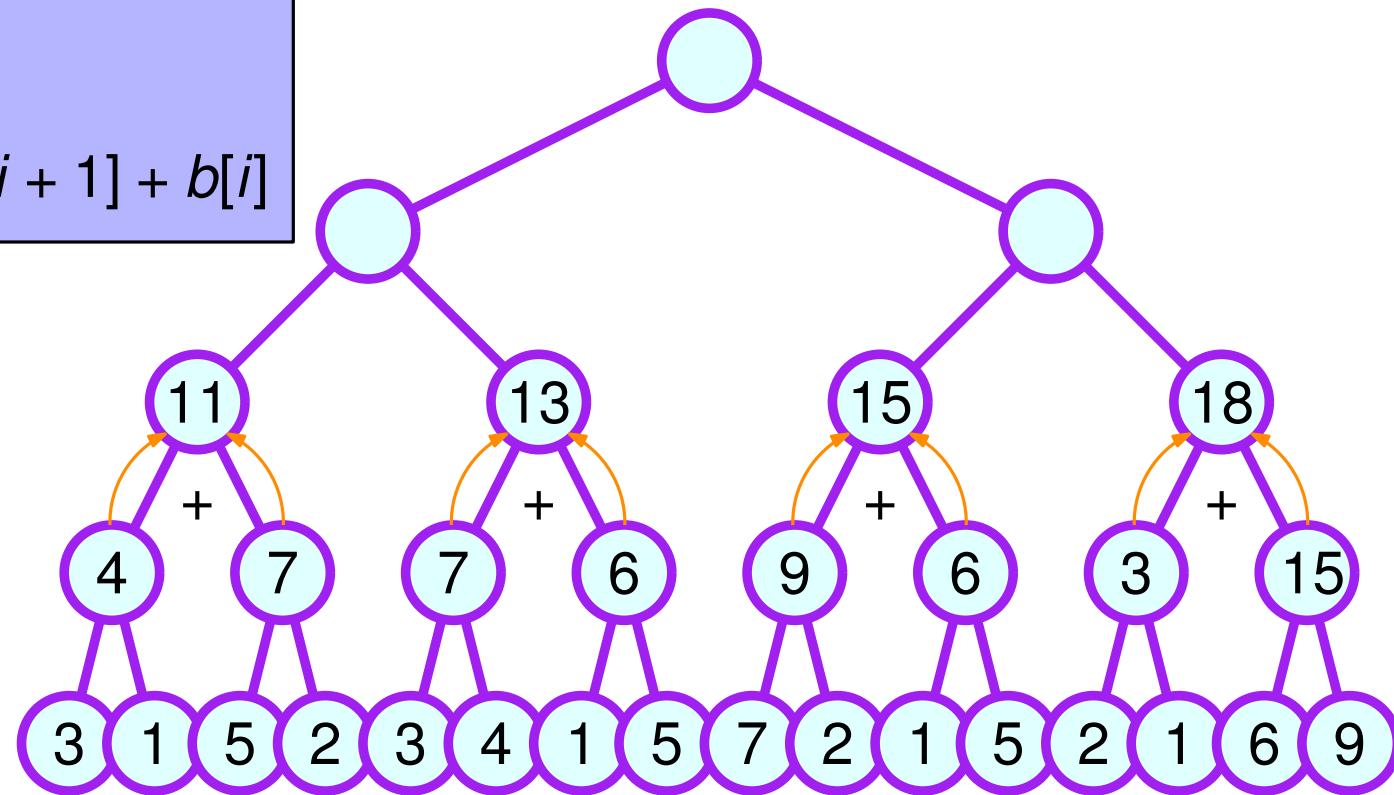
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

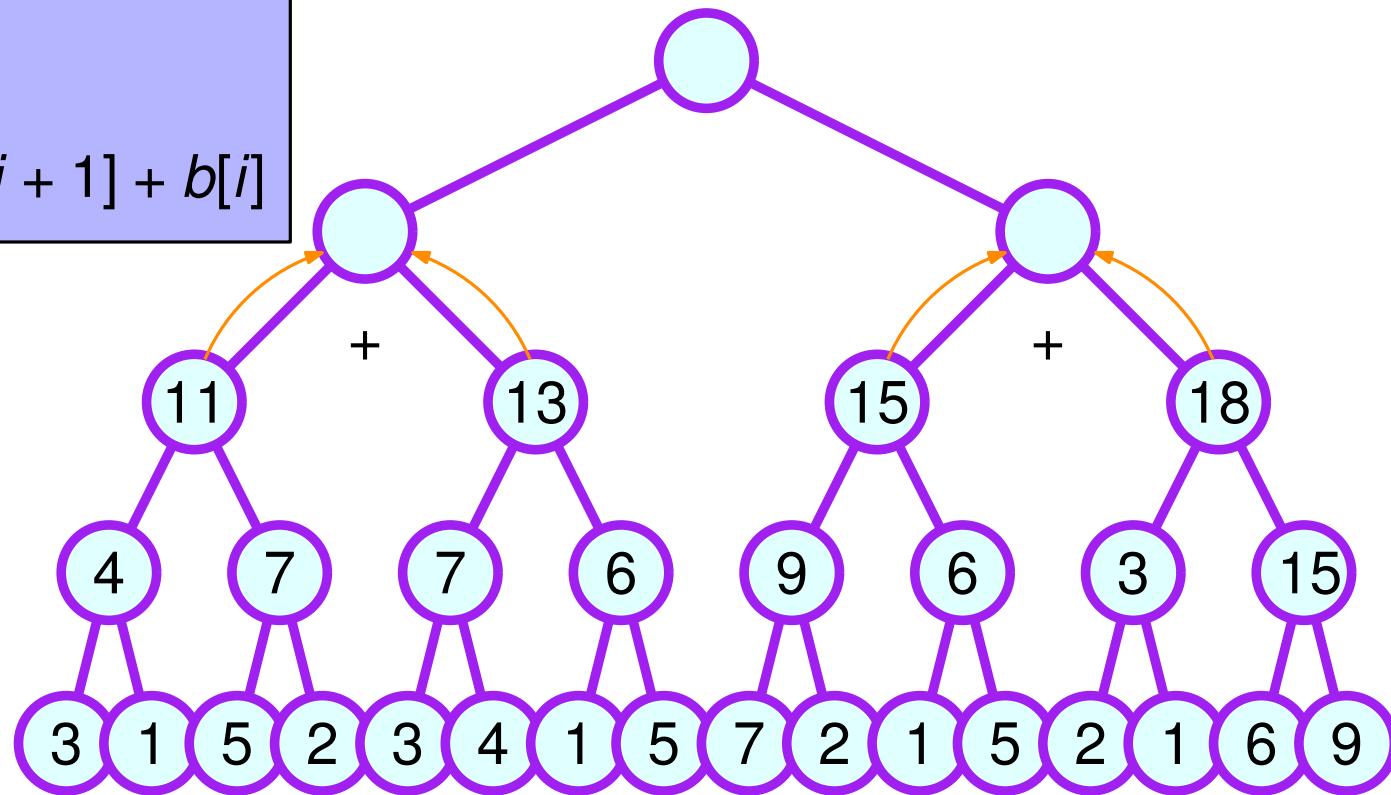
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

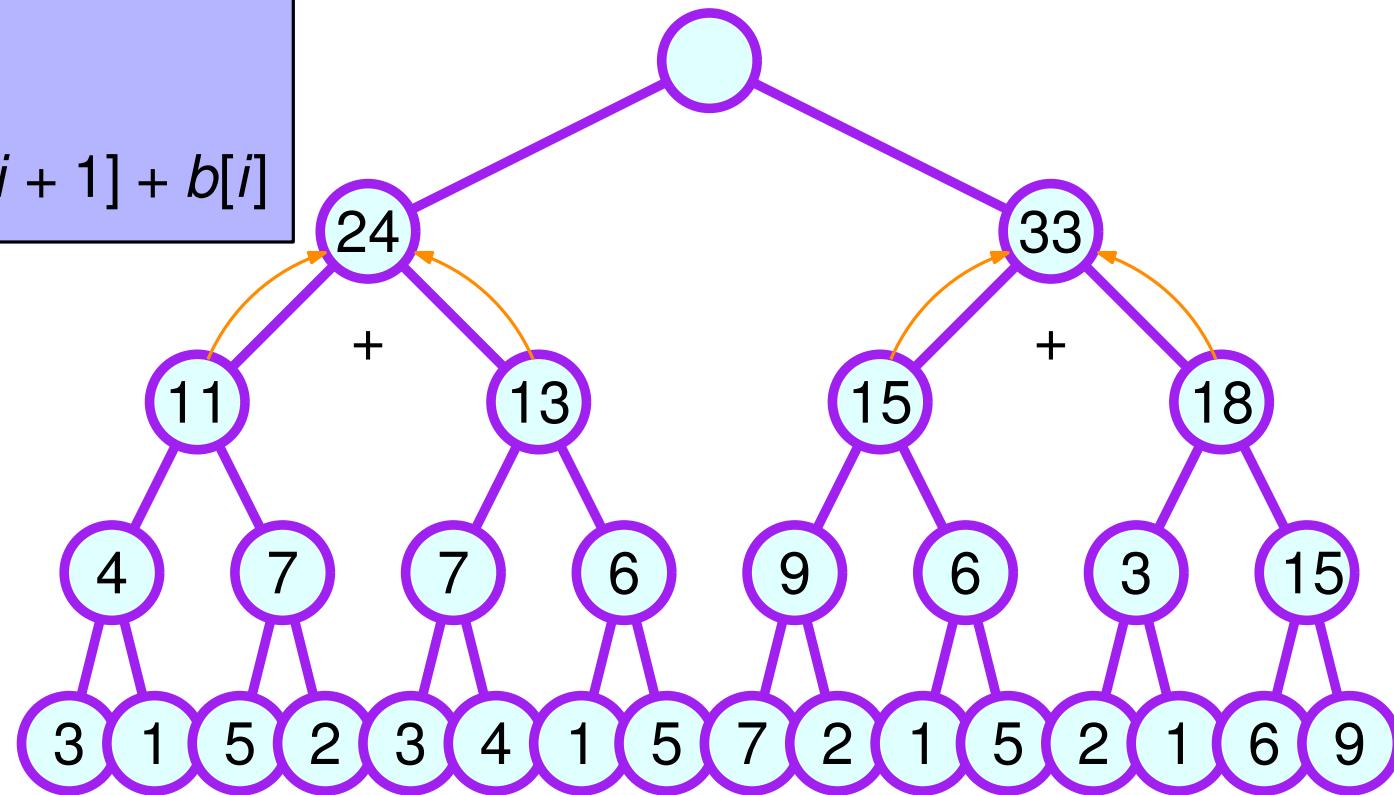
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

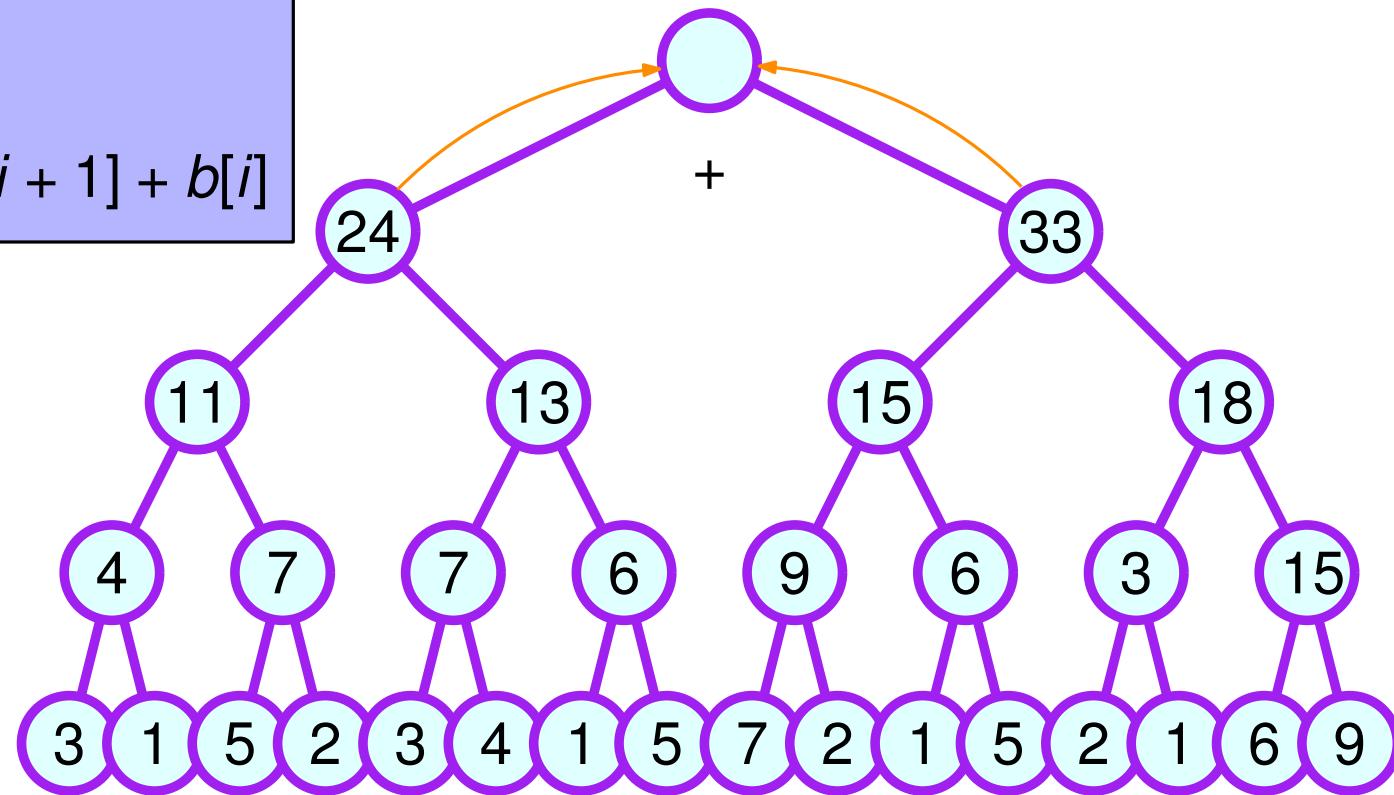
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

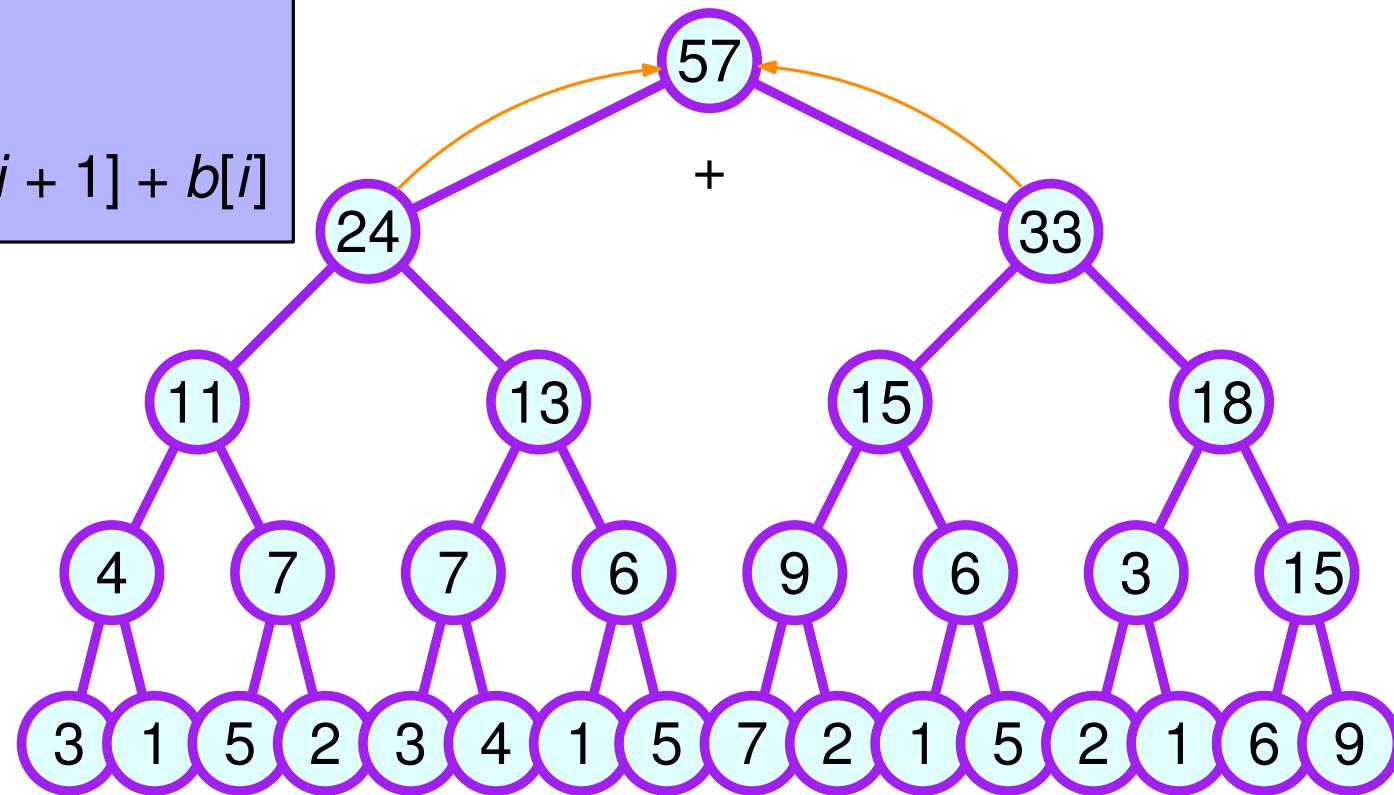
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

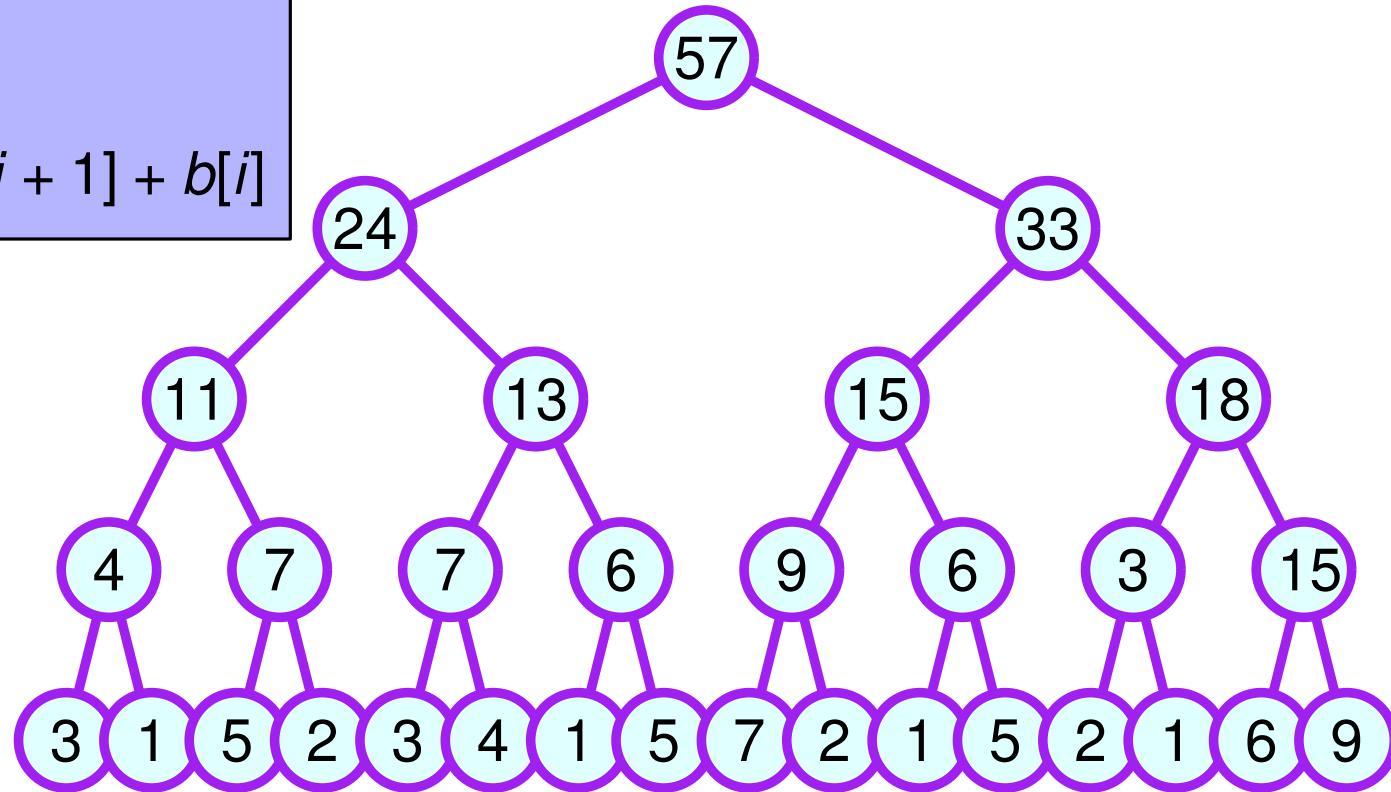
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

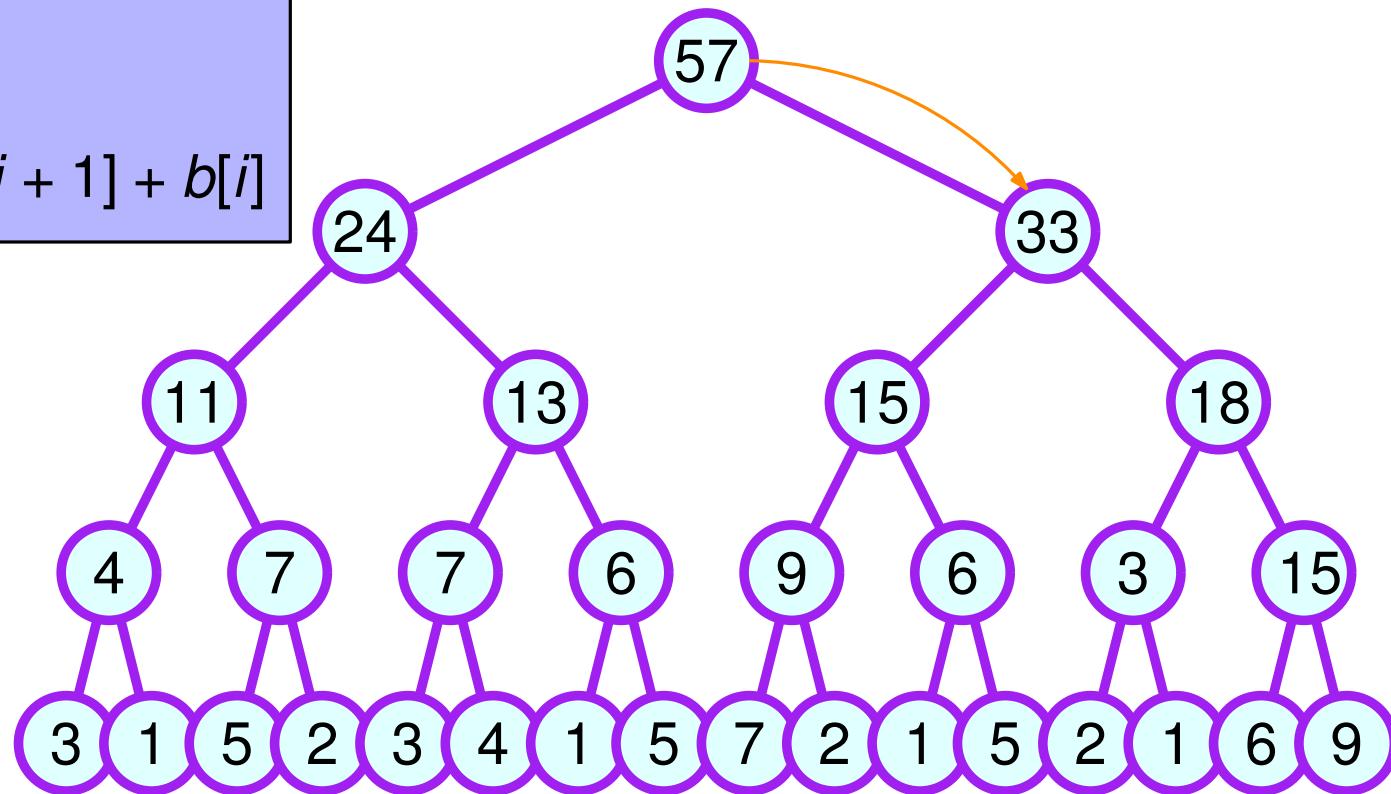
Up-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

Down-sweep

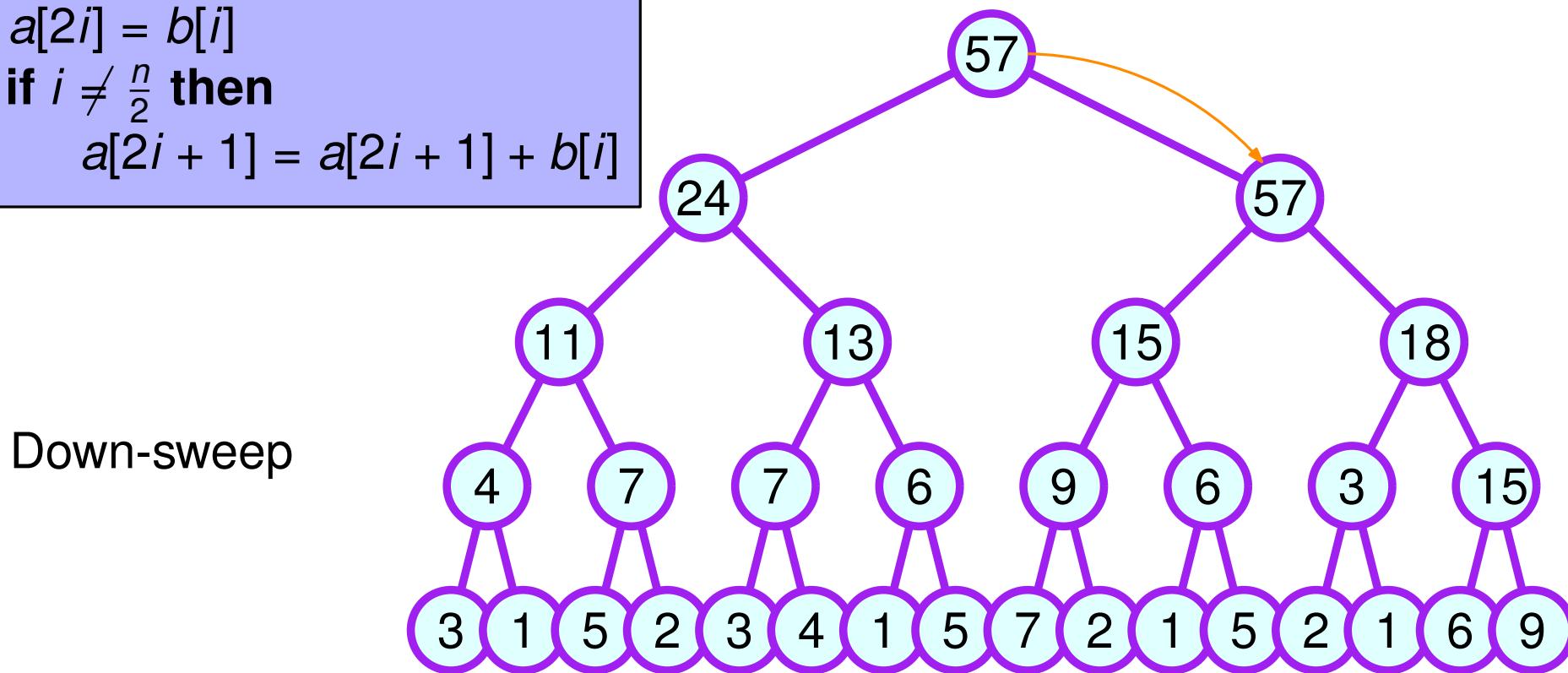


Balanced-Tree Technique

```

procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 

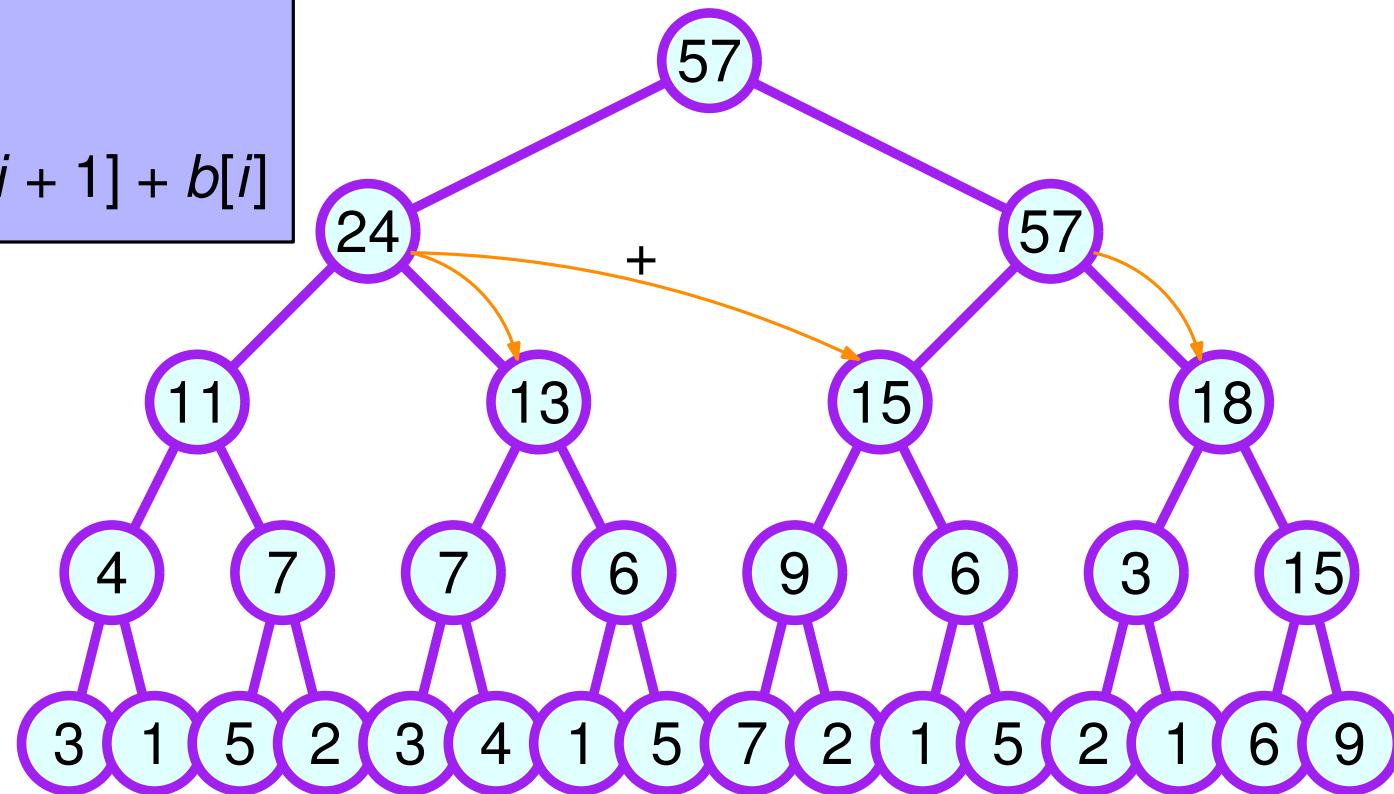
```



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

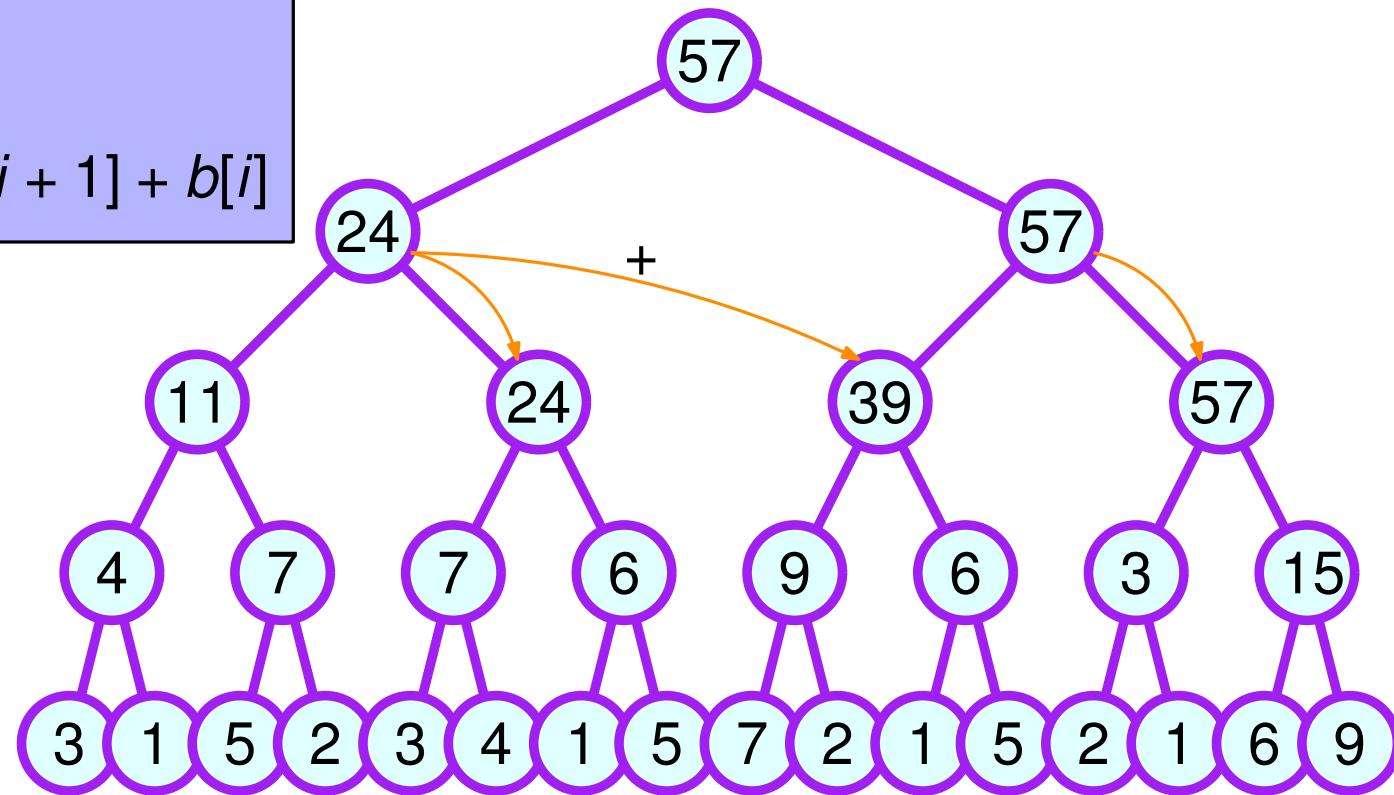
Down-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

Down-sweep

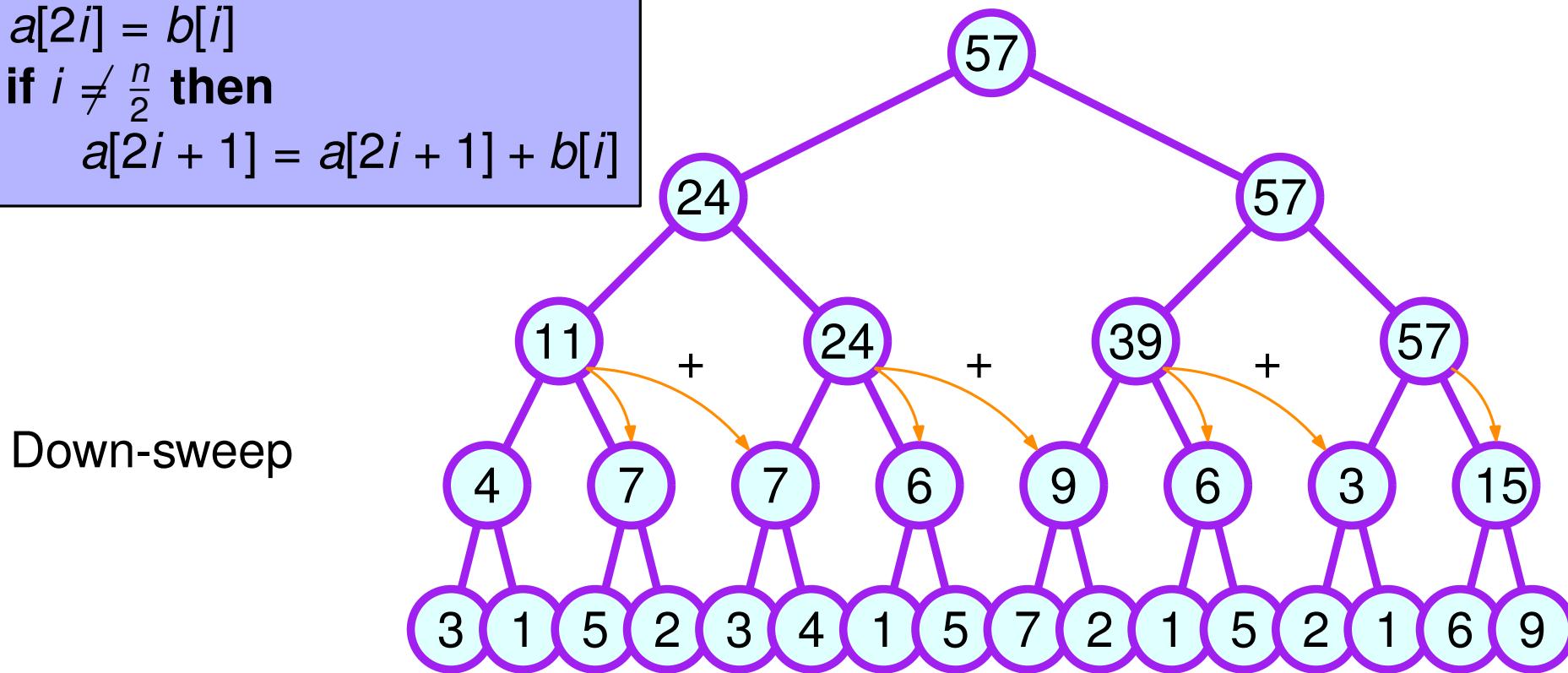


Balanced-Tree Technique

```

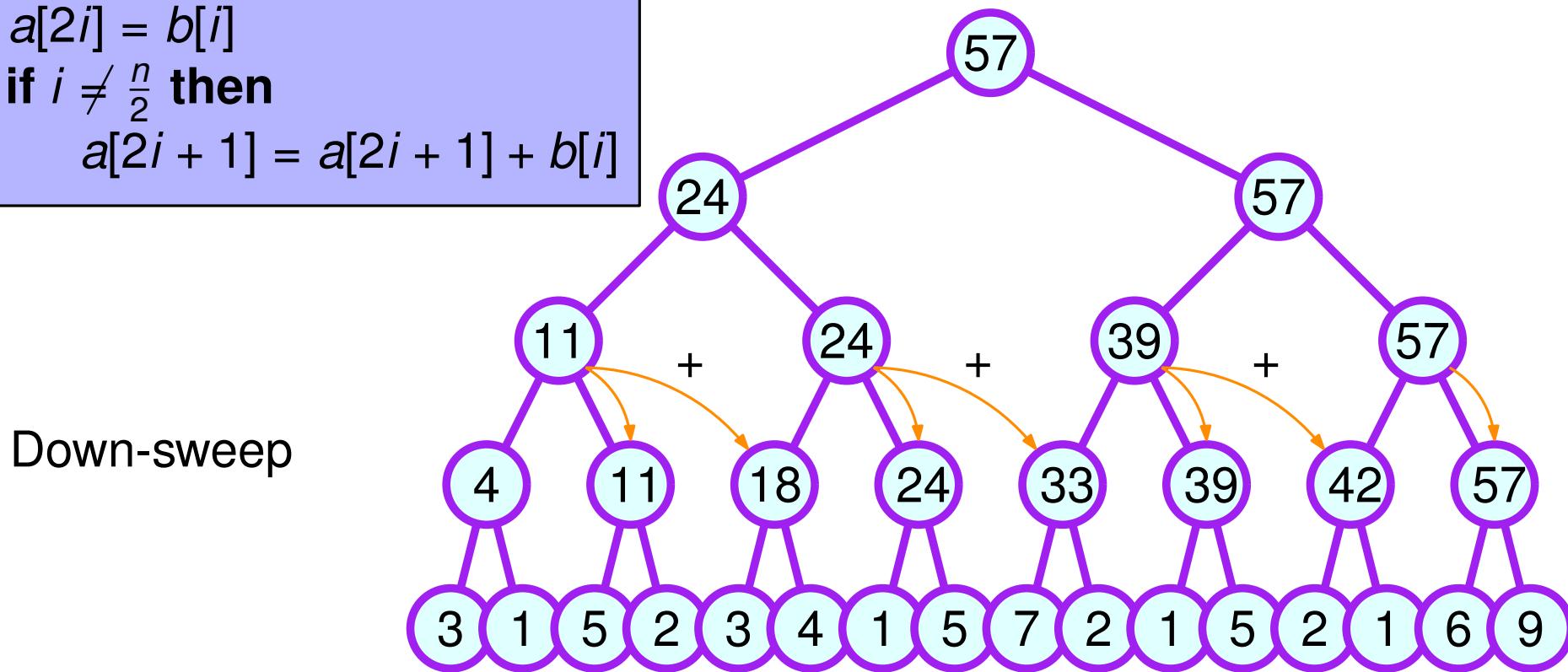
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 

```



Balanced-Tree Technique

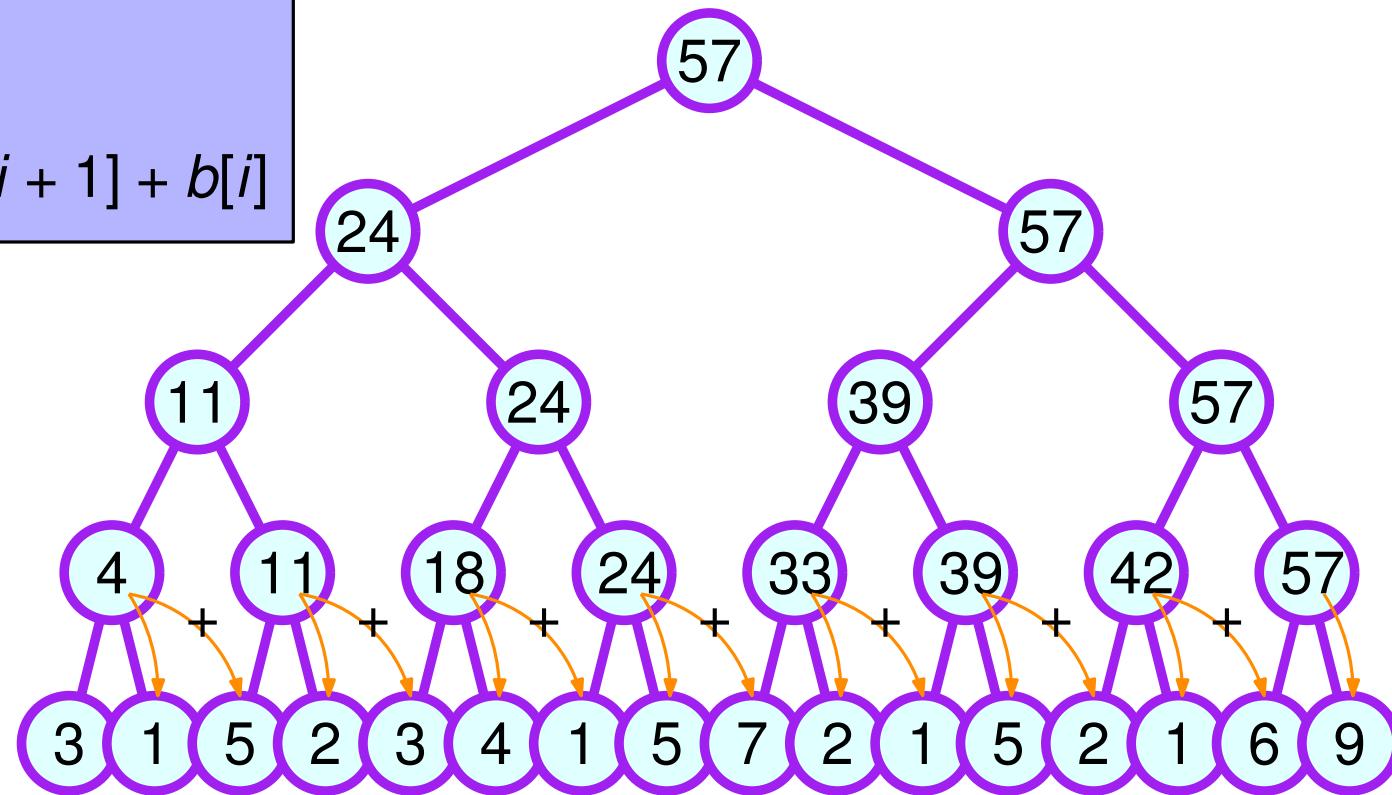
```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

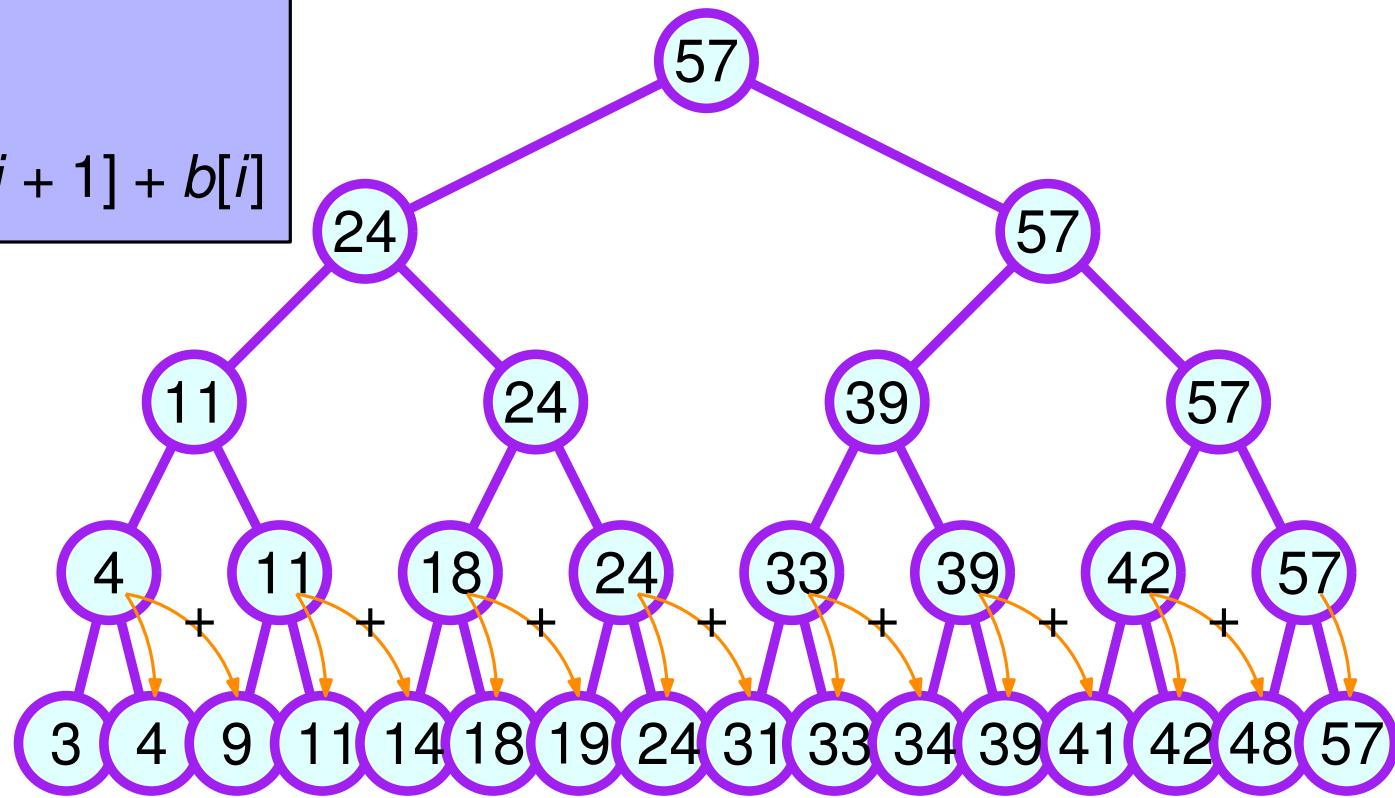
Down-sweep



Balanced-Tree Technique

```
procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 
```

Down-sweep

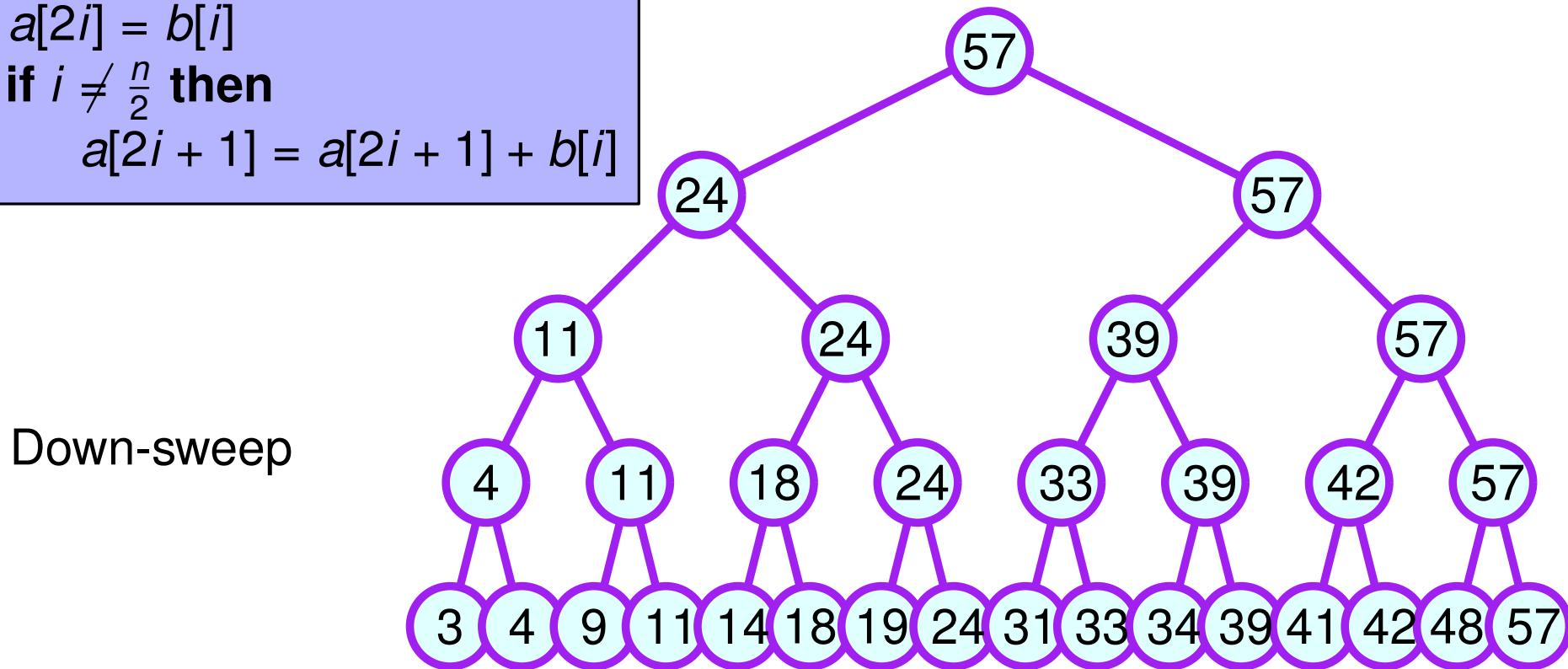


Balanced-Tree Technique

```

procedure PREFIX-SUMS( $a[1..n]$ )
  if  $n \leq 1$  then return
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $b[i] = a[2i - 1] + a[2i]$ 
    PREFIX-SUMS( $b[1..\frac{n}{2}]$ )
  for  $i = 1$  to  $\frac{n}{2}$  in parallel do
     $a[2i] = b[i]$ 
    if  $i \neq \frac{n}{2}$  then
       $a[2i + 1] = a[2i + 1] + b[i]$ 

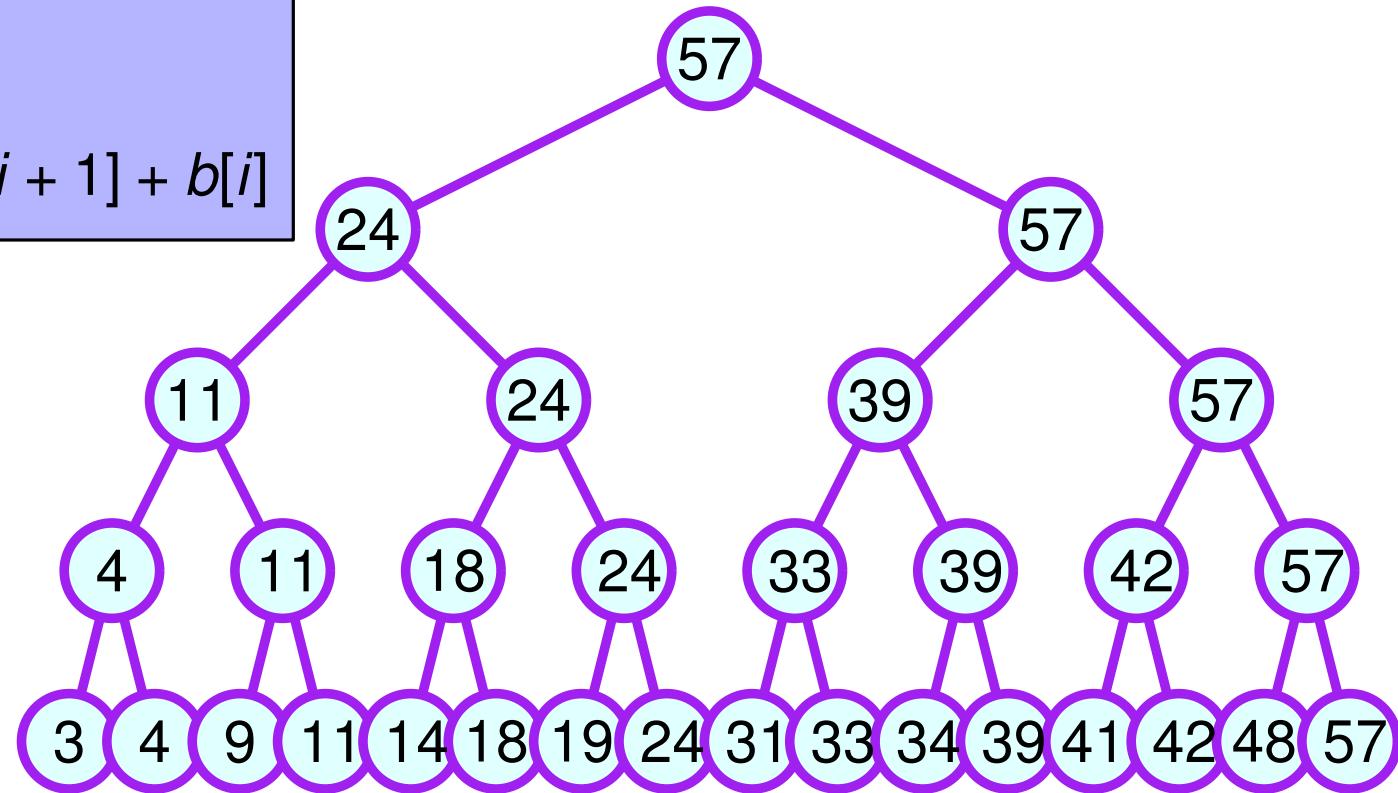
```



Balanced-Tree Technique

```
procedure PREFIX-SUMS(a[1..n])
  if n ≤ 1 then return
  for i = 1 to  $\frac{n}{2}$  in parallel do
    b[i] = a[2i - 1] + a[2i]
    PREFIX-SUMS(b[1.. $\frac{n}{2}$ ])
  for i = 1 to  $\frac{n}{2}$  in parallel do
    a[2i] = b[i]
    if i ≠  $\frac{n}{2}$  then
      a[2i + 1] = a[2i + 1] + b[i]
```

Works with any
associative operation



Balanced-Tree Technique

```
procedure PREFIX-SUMS(a[1..n])
  if n ≤ 1 then return
  for i = 1 to  $\frac{n}{2}$  in parallel do
    b[i] = a[2i - 1] + a[2i]
    PREFIX-SUMS(b[1.. $\frac{n}{2}$ ])
  for i = 1 to  $\frac{n}{2}$  in parallel do
    a[2i] = b[i]
    if i ≠  $\frac{n}{2}$  then
      a[2i + 1] = a[2i + 1] + b[i]
```

Works with any
associative operation

$$(a + b) + c = a + (b + c)$$

$$(a \cdot b) \cdot c = a \cdot (b \cdot c)$$

$$\min(\min(a, b), c) = \min(a, \min(b, c))$$

