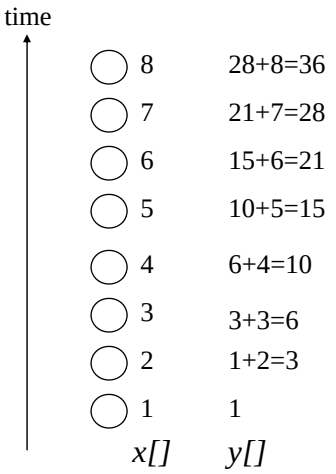
The Prefix-sum algorithm

- Given *x[0]+…+x[n-1]*, compute *y[0] , … , y[n-1]* such that:

$$y[i] = \sum_{j=0}^{i} x[j]$$

```
y[0] = x[0]
For (i = 1 ; i < 8  ;  i++)
  {
    y[i] = y[i-1] + x[i]
  }
```

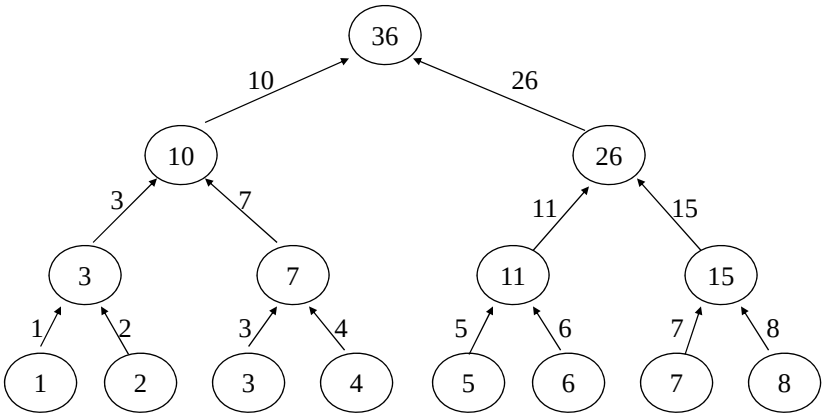
x[i] = i+1



Parallel prefix-sum algorithm

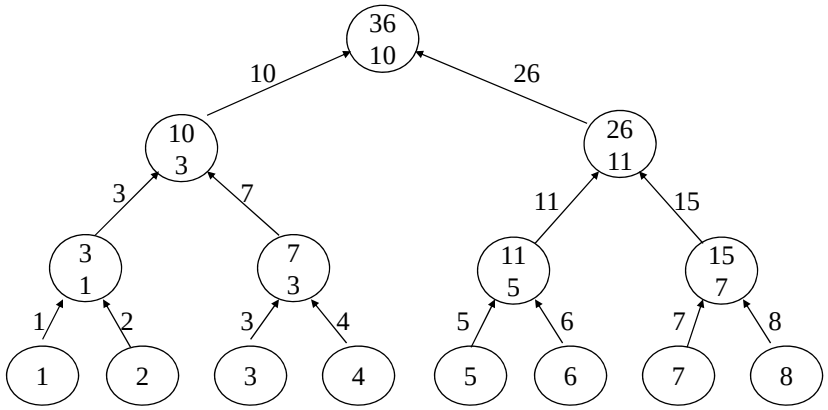
Use a tree similar to the one used to compute the sum.

1) Upward phase: proceed as in the case of computing the global sum (*log n* steps).



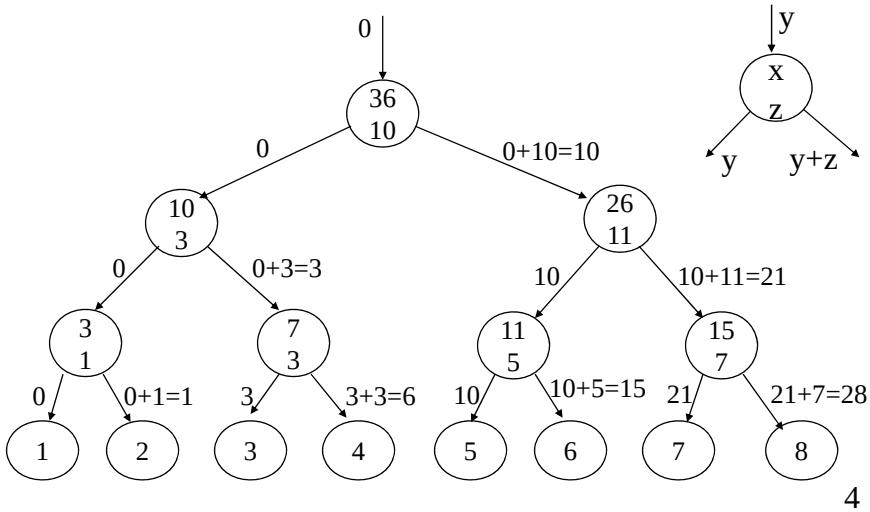
Parallel prefix-sum algorithm

2) During the upward phase, store at each node, n, the sum of the leaves in the left sub-tree of n (call this sum “z”).



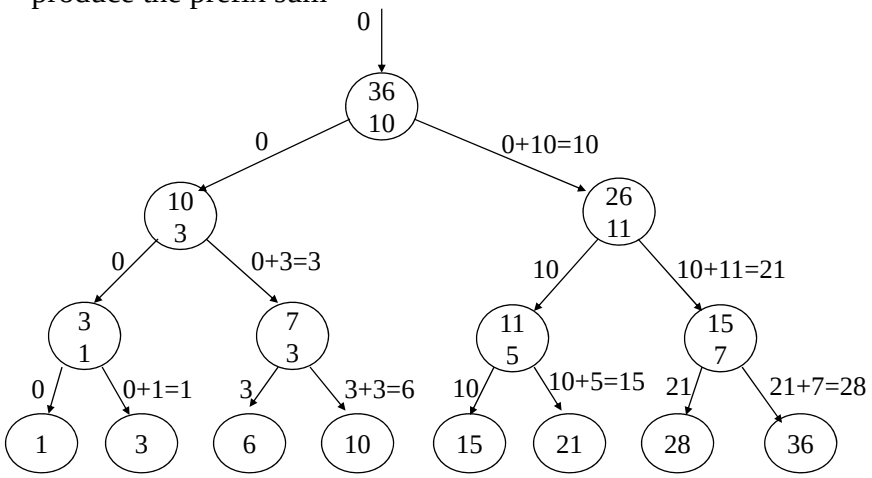
Parallel prefix-sum algorithm

3) Downward phase (*log n* steps): each node receive “y” from parent (0 if root), sends y to left child and “y+z” to right child.



Parallel prefix-sum algorithm

4) Leaves add the value received from parent to its own value to produce the prefix sum



Parallel prefix-sum algorithm