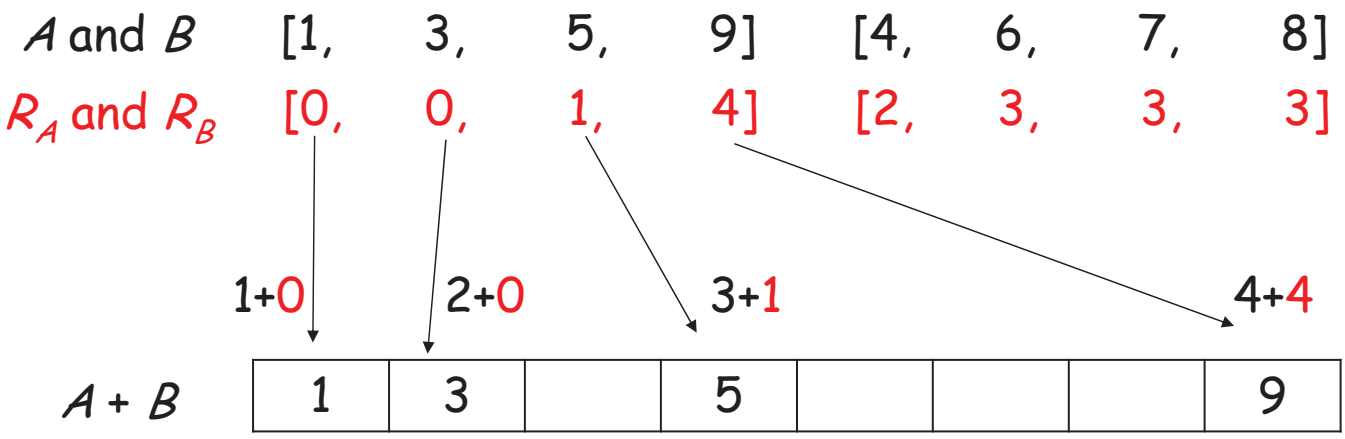
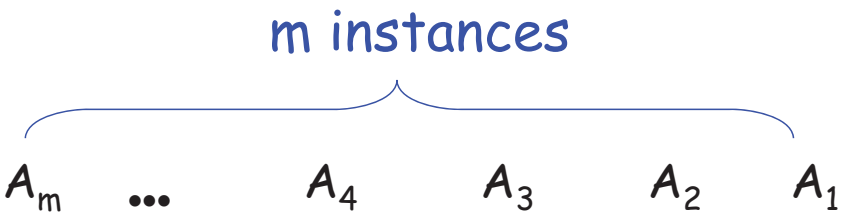
Merge - Sequential Algorithm: $O(n)$ time

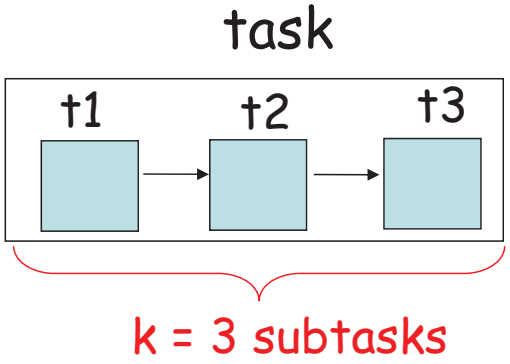
A	B	
[1, 3, 5, 9]	[4, 6, 7, 8]	$\Rightarrow 1$
[1, 3, 5, 9]	[4, 6, 7, 8]	$\Rightarrow 3$
[1, 3, 5, 9]	[4, 6, 7, 8]	$\Rightarrow 4$
[1, 3, 5, 9]	[4, 6, 7, 8]	$\Rightarrow 5$



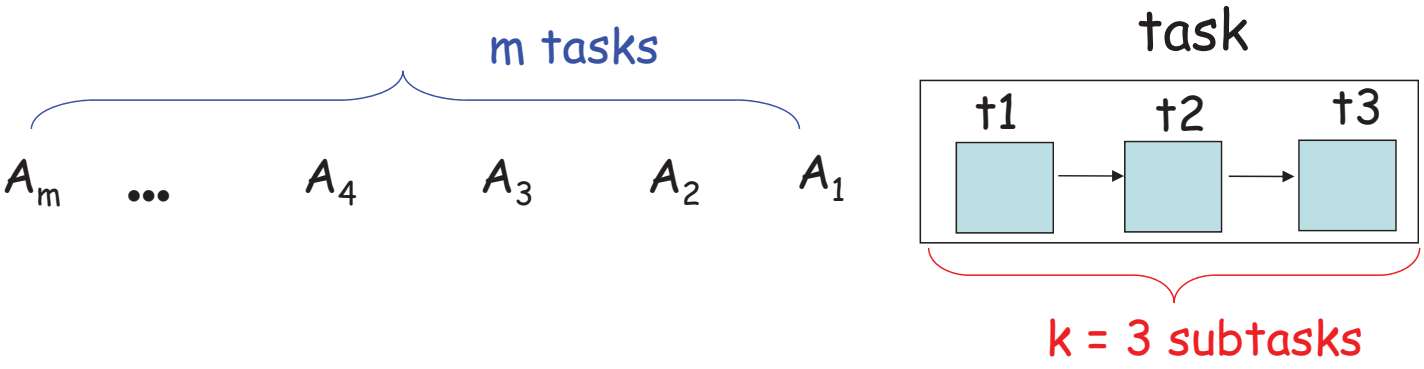
Ranking of a subtask
- Sequential Algorithm: $O(n)$ time

	5	6	7	8		15	16	17	18	
A_2	[4,	6,	8,	10]	B_2	[7,	9,	11,	13]	
	↑					↑				⇒ 4
	rank = 14									
	[4,	6,	8,	10]		[7,	9,	11,	13]	
		↑				↑				⇒ 6
		rank = 14								
	[4,	6,	8,	10]		[7,	9,	11,	13]	
			↑			↑				⇒ 7
						rank = 6				
	[4,	6,	8,	10]		[7,	9,	11,	13]	
			↑				↑			⇒ 8
			rank = 15							





Time: $3 \times m$ or $k \times m$

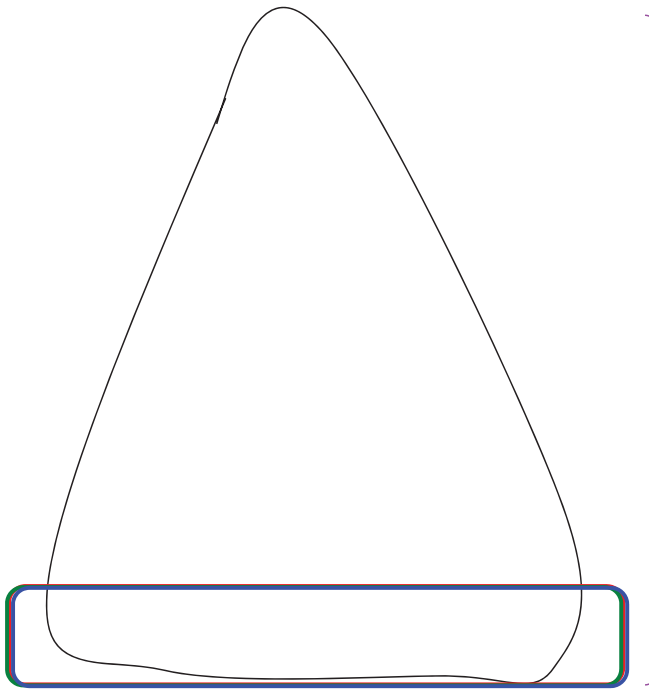


Time: $3 \times m$ or $k \times m$

$\Downarrow$

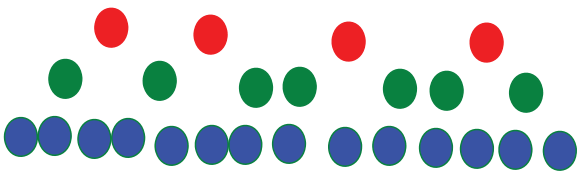
$3 + (m-1)$ or $k + (m-1)$

speedup = k

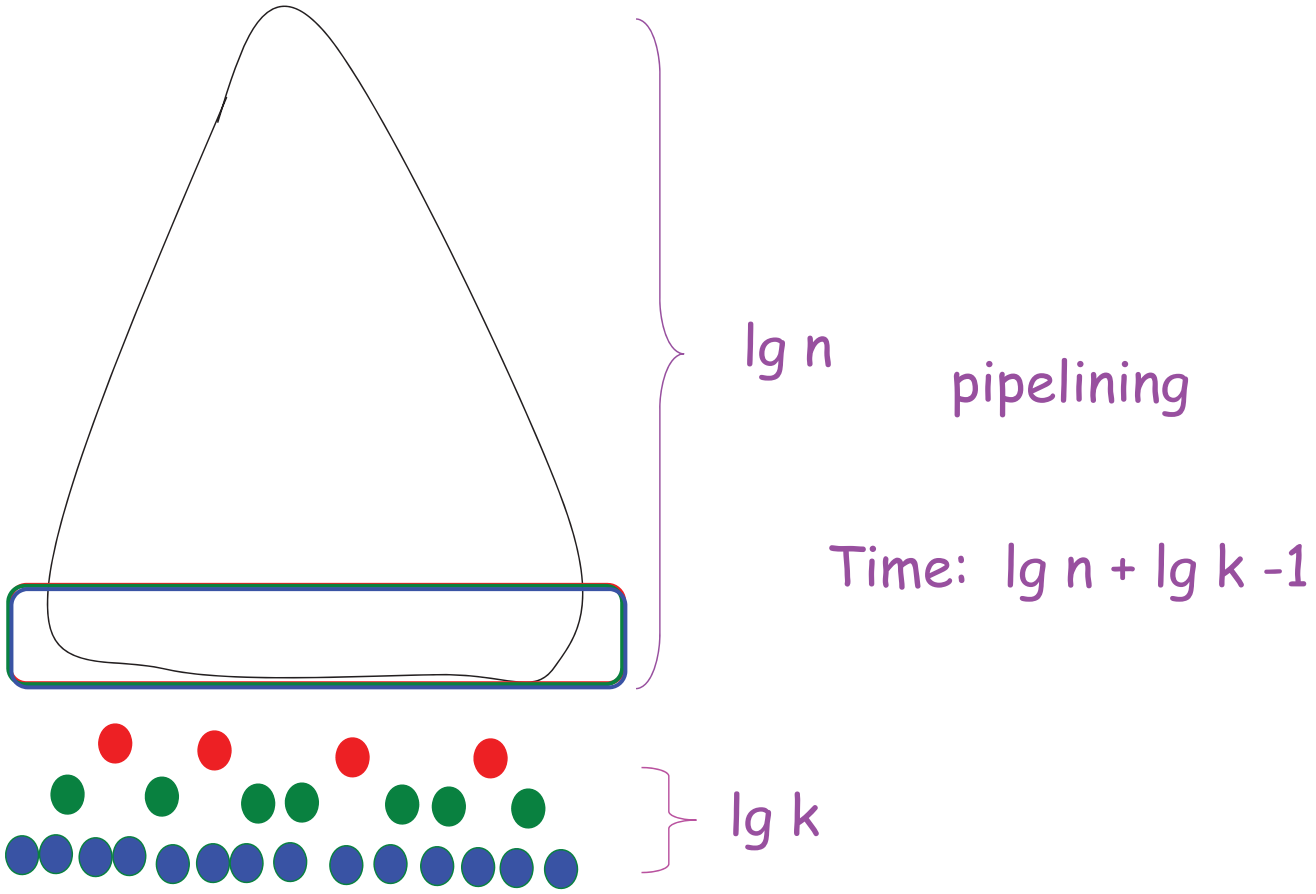


lg n

Time: $\lg k \times \lg n$



lg k



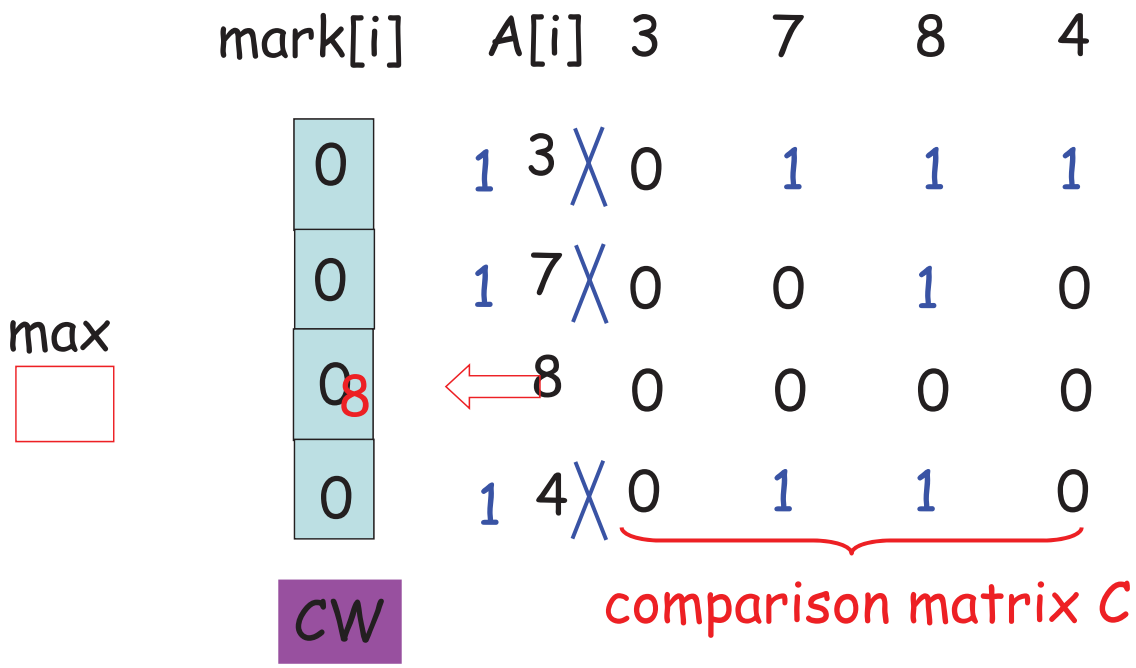
Using pipelining

Chain: lg k * lg n ===> lg n + lg k -1

Implementation

How to know who are the medians ? (CR, ER)

How to insert 3 keys to a node at the same time?



Finding maximum on the CRCW PRAM

Algo 1: 1 PE, O(n) time, O(n) cost (slow but cheap)

Algo 2: n PE, O(lglg n) time, O(n lglg n) cost (fast but expensive)

Cascading: using p PEs (Example: p = 16)

Stage 1: Algo 1 - O(n/p) time

Stage 2: Algo 2 - O(lglg p) time