



Merge algorithm

2	3	4	5	7
---	---	---	---	---

1	2	4	4	6
---	---	---	---	---

Take two **sorted** lists
merge them
Into one **sorted** list

1	2	2	3	4	4	4	5	6	7
---	---	---	---	---	---	---	---	---	---

Merge algorithm

2	3	4	5	7
---	---	---	---	---

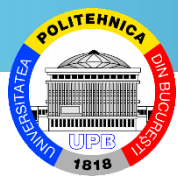
1	2	4	4	6
---	---	---	---	---

Solution:

Always extract the smallest element
from the lists
(guaranteed to be first in one of them)

$O(n)$ complexity

1	2	2	3	4	4	4	5	6	7
---	---	---	---	---	---	---	---	---	---



Merge algorithm

2	3	4	5	7
1	2	4	4	6



Merge algorithm

2	3	4	5	7
	2	4	4	6

1



Merge algorithm

3	4	5	7
2	4	4	6

1	2
---	---



Merge algorithm

3	4	5	7
	4	4	6

1	2	2
---	---	---



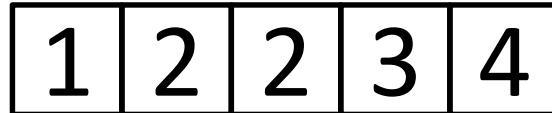
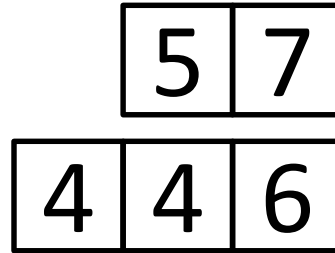
Merge algorithm

4	5	7
4	4	6

1	2	2	3
---	---	---	---



Merge algorithm





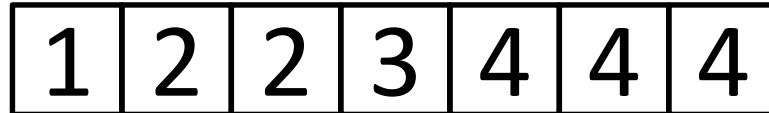
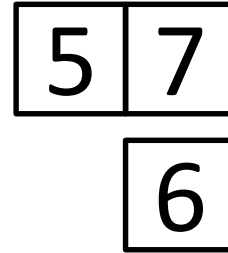
Merge algorithm

5	7
4	6

1	2	2	3	4	4
---	---	---	---	---	---



Merge algorithm





Merge algorithm

7

6

1	2	2	3	4	4	4	5
---	---	---	---	---	---	---	---



Merge algorithm

7

1	2	2	3	4	4	4	5	6
---	---	---	---	---	---	---	---	---



Merge algorithm

1	2	2	3	4	4	4	5	6	7
---	---	---	---	---	---	---	---	---	---

Merge algorithm

1	2	2	3	4	4	4	5	6	7
---	---	---	---	---	---	---	---	---	---

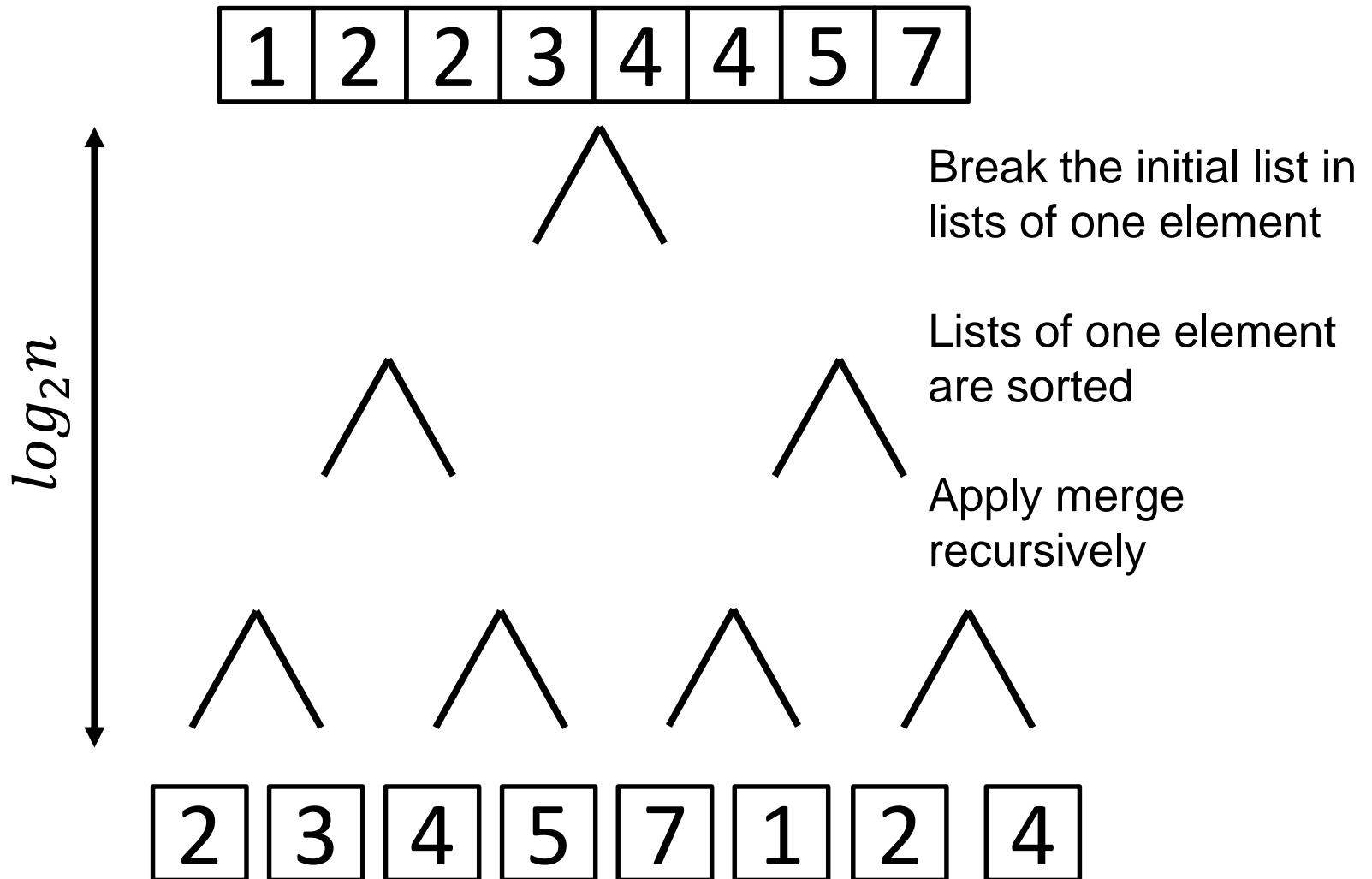
For next slides this
sign means merge



2	3	4	5	7
---	---	---	---	---

1	2	4	4	6
---	---	---	---	---

Merge sort



Merge sort



Sequential complexity:

.....

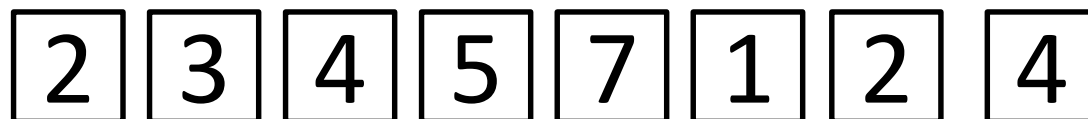
$$O(n * \log_2 n)$$



.....

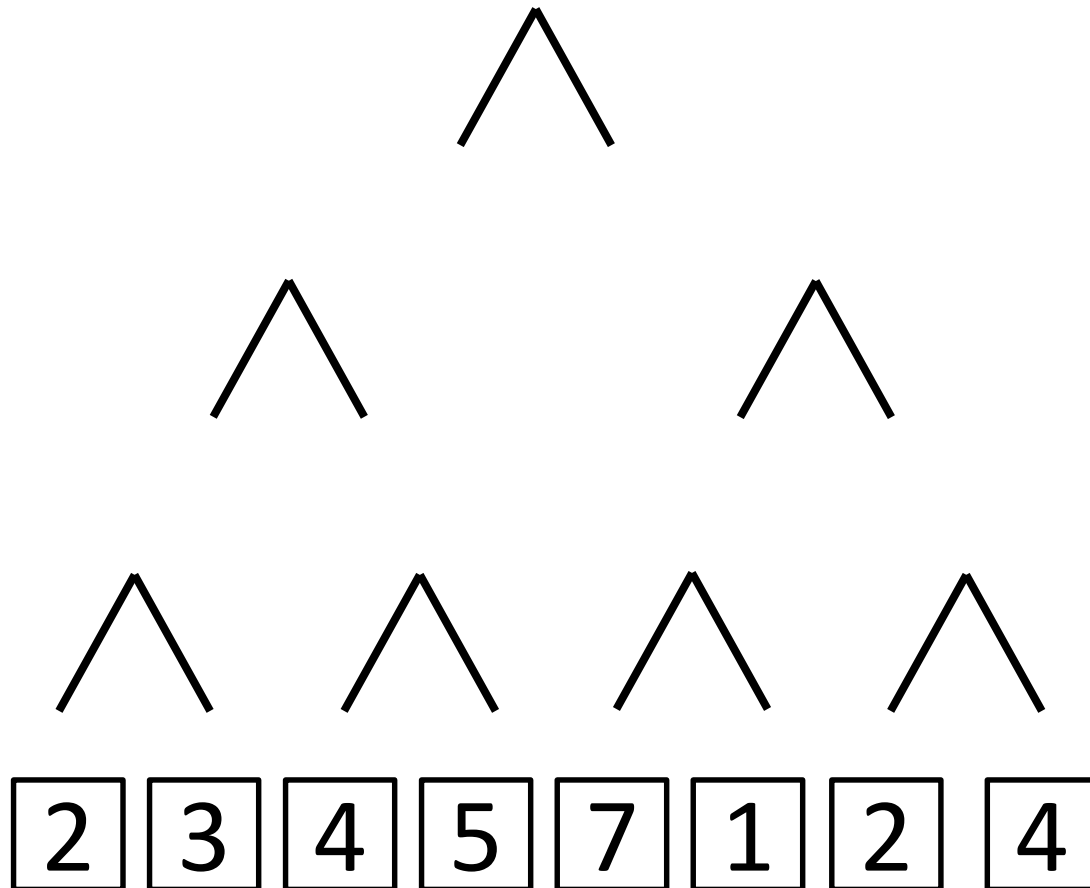


.....

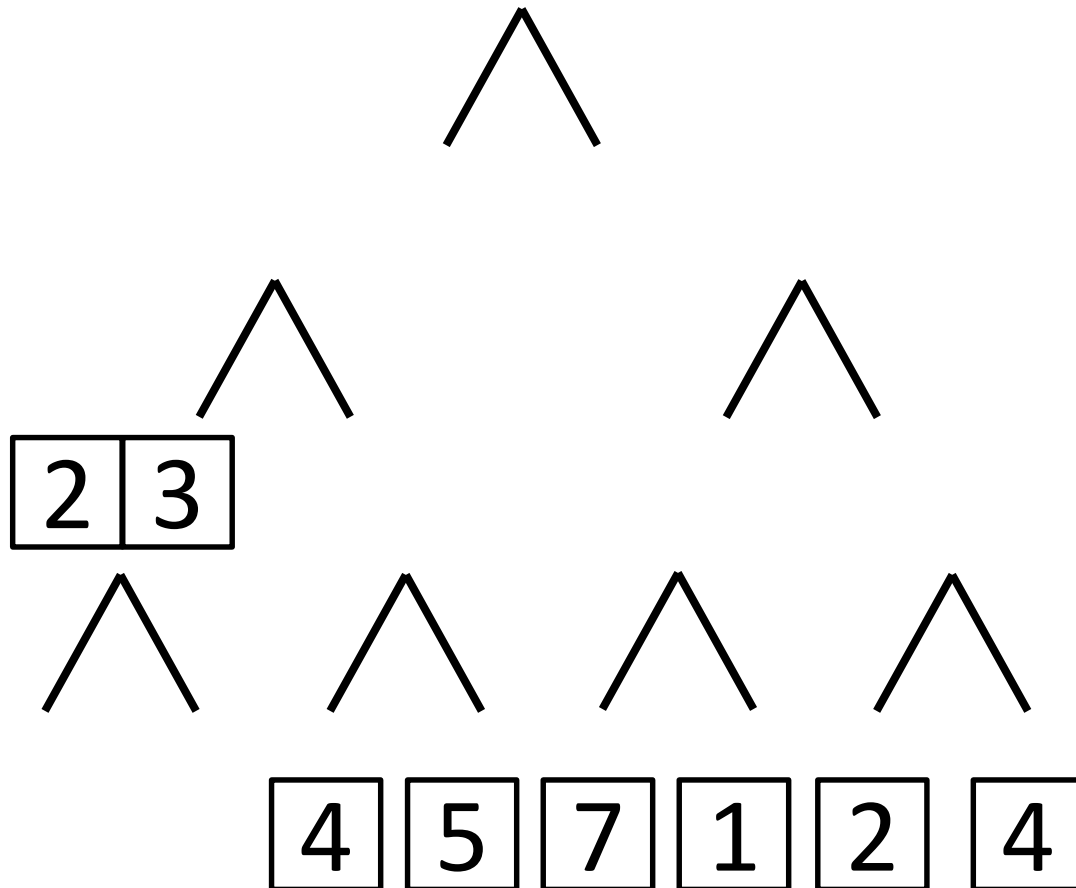


.....

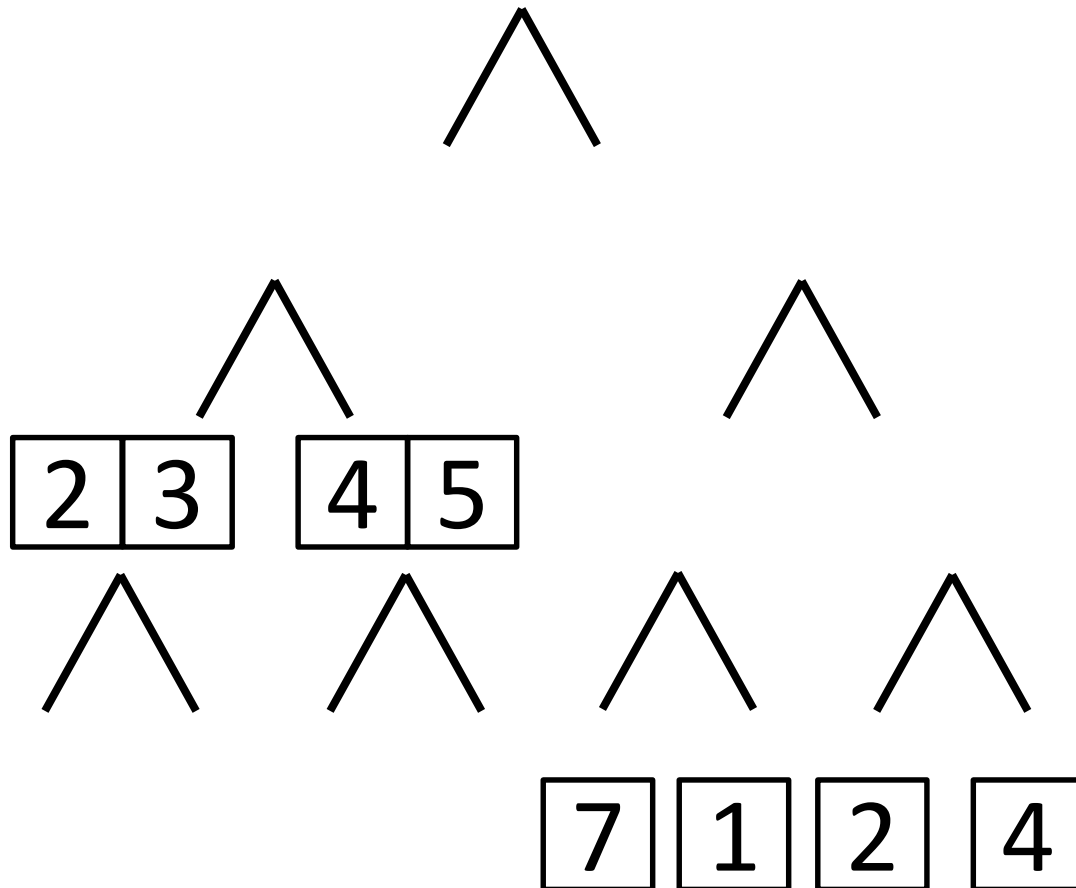
Merge sort



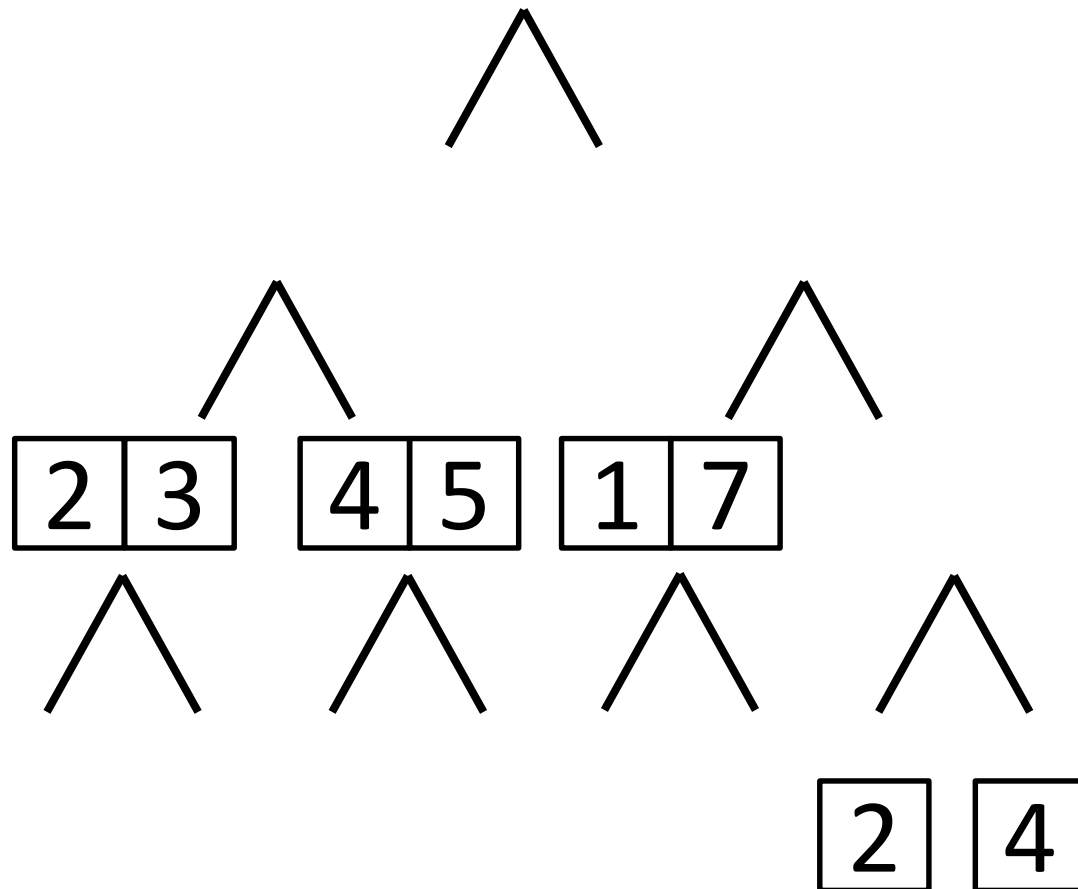
Merge sort



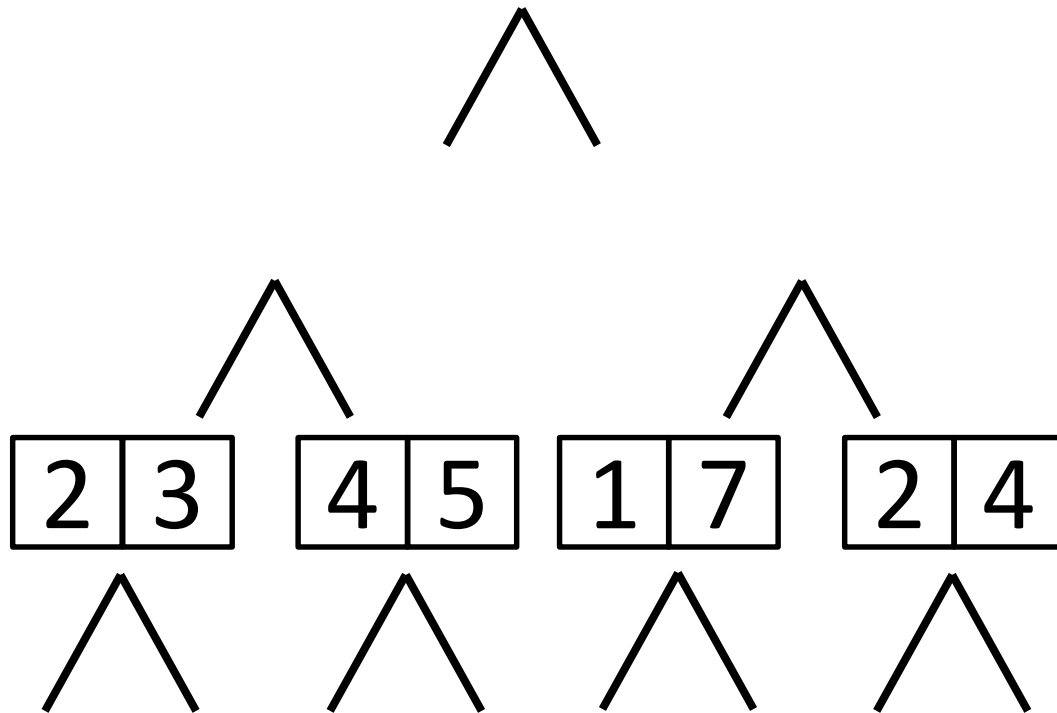
Merge sort



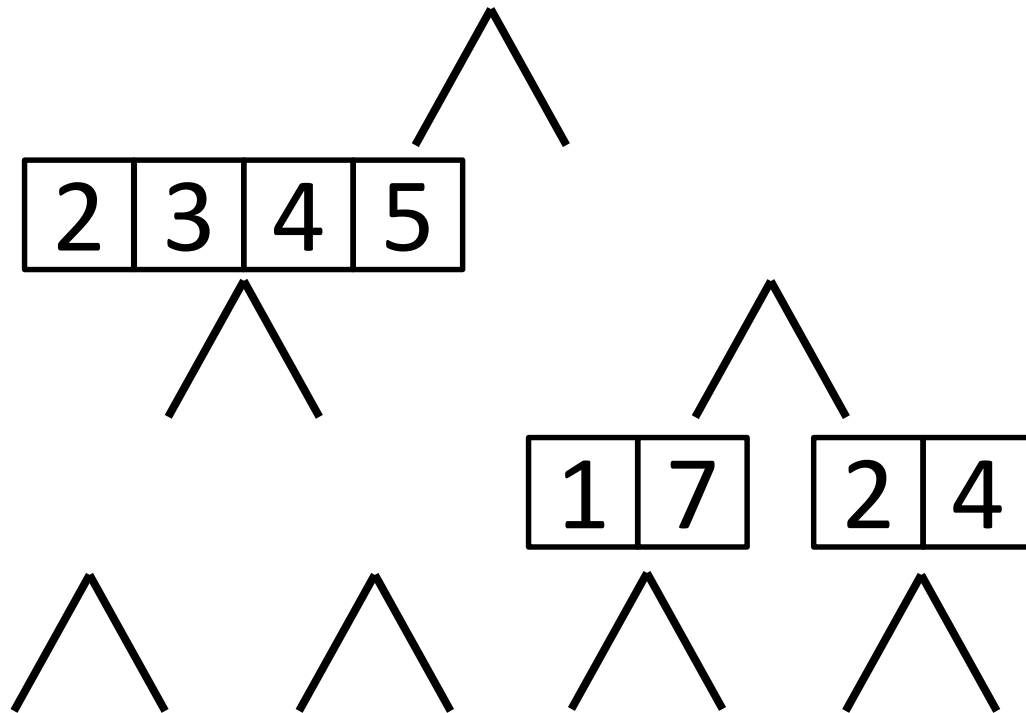
Merge sort

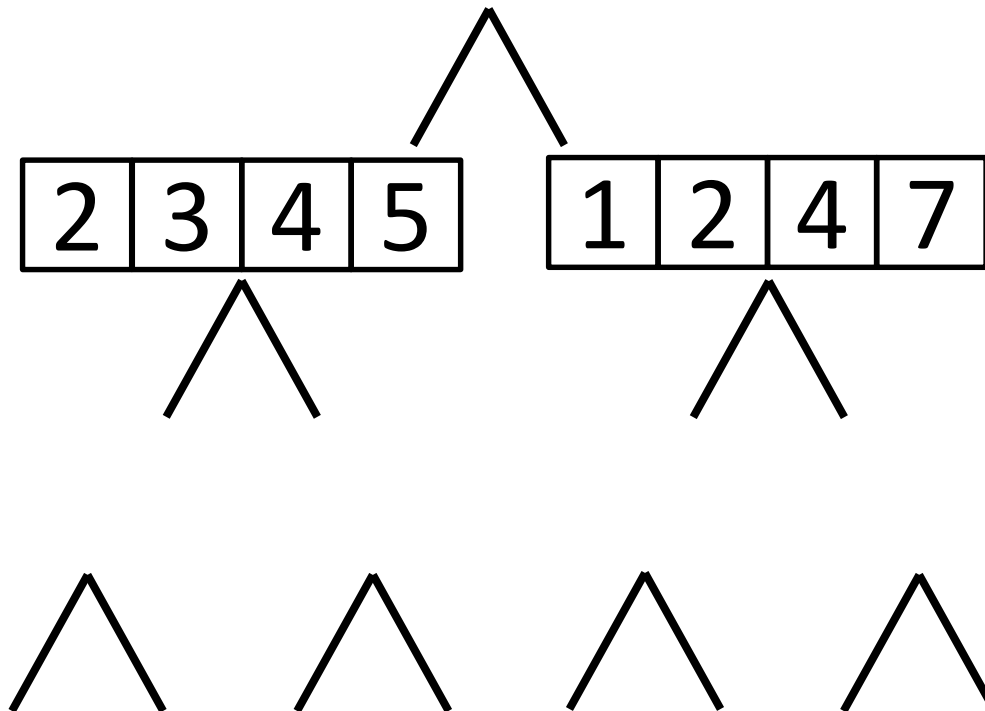


Merge sort



Merge sort



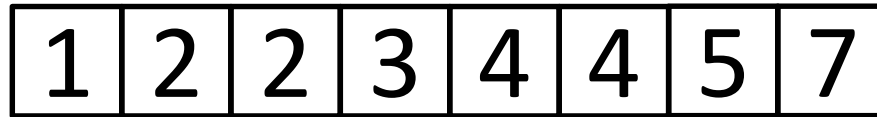


Merge sort

1	2	2	3	4	4	5	7
---	---	---	---	---	---	---	---



Parallel merge sort



Can be
executed
in parallel →



Can be
executed
in parallel →



Parallel merge sort



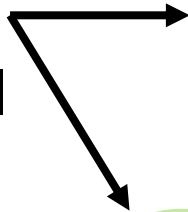
Result of



depends on
result of



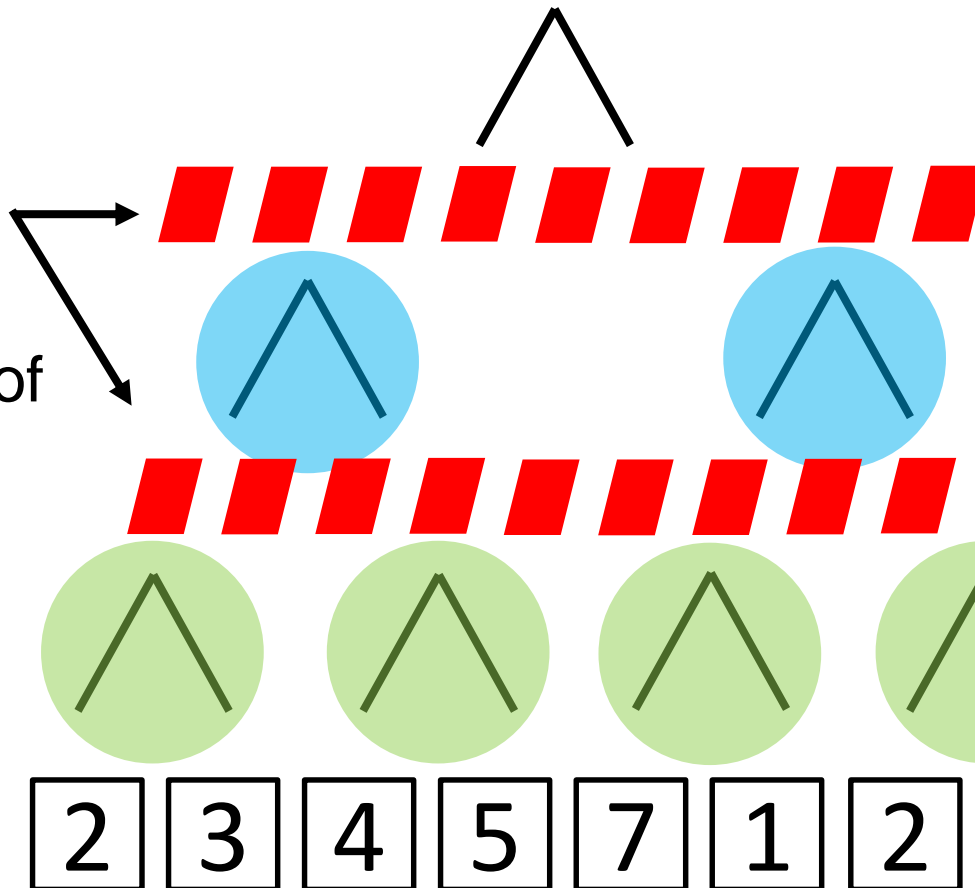
Can **NOT** be
executed
in parallel



Parallel merge sort



Solution:
Barrier
between
each level of
the tree



Each level of
the tree
needs to wait
for the
operations on
the previous
level to finish

Parallel merge sort

Parallel complexity:

p = processor cores

for p=n

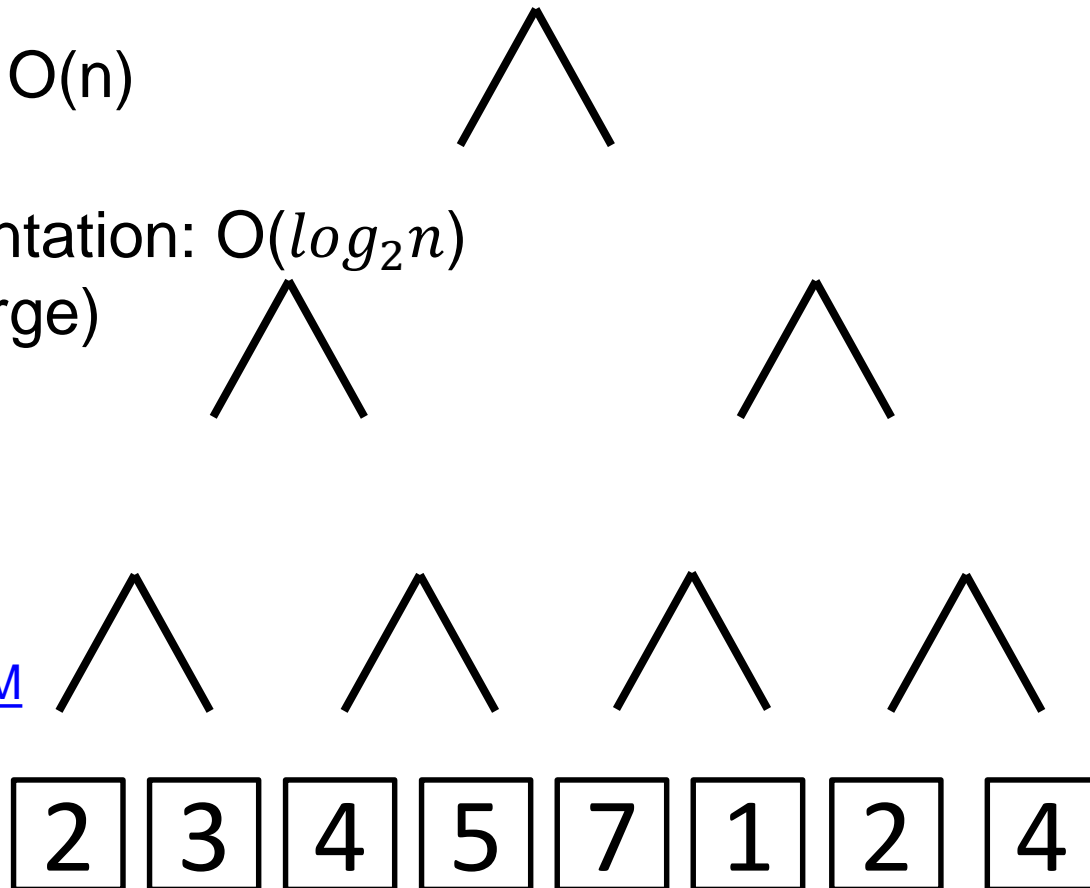
$$O(\sum_{i=1}^{\log_2 n} 2^i) = O(n)$$

Best implementation: $O(\log_2 n)$

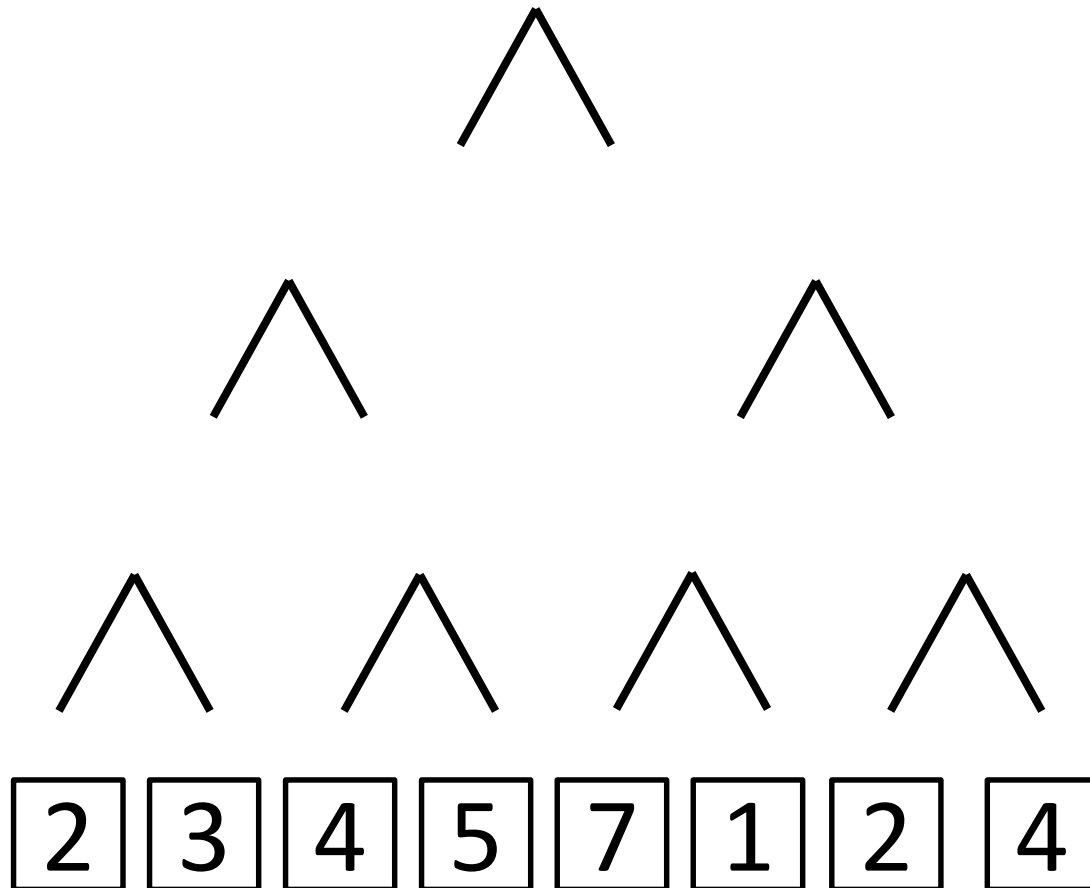
(with $O(1)$ merge)



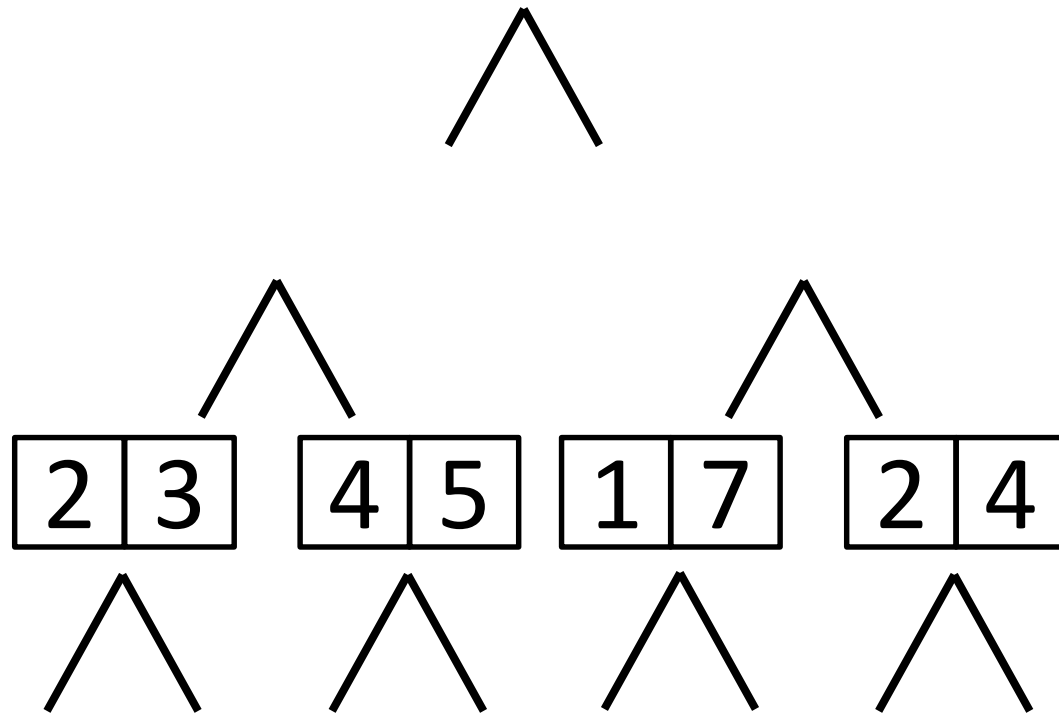
<http://goo.gl/okU3fM>



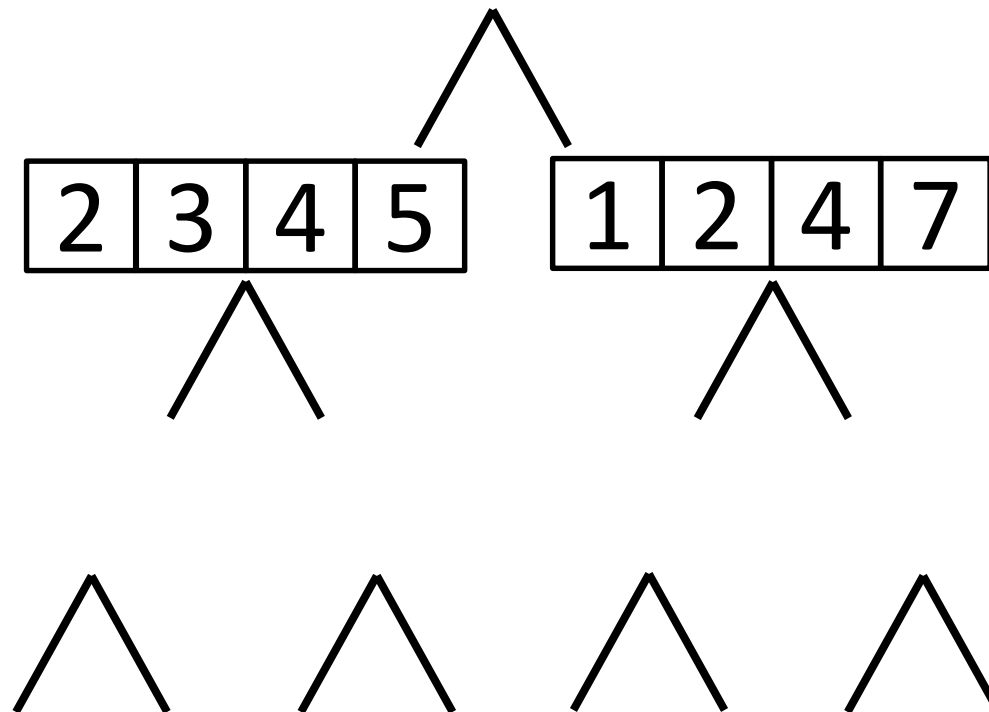
Parallel merge sort



Parallel merge sort



Parallel merge sort



Parallel merge sort

1	2	2	3	4	4	5	7
---	---	---	---	---	---	---	---

