Divide and Conquer

Class 16

Divide and Conquer

- binary search is an example of divide and conquer
- an algorithm design strategy
- basis for several famous efficient algorithms

1. partition current problem instance into non-overlapping smaller problem instances
2. solve smaller instances separately (often recursively)
3. combine small instance solutions into larger instance solution (not always necessary, depending on problem)

Mergesort

- a very popular sorting algorithm
- directly uses the divide-and-conquer pattern

Merge

- mergesort requires the merge operation
- two input arrays a and b, each already sorted, with m and n elements respectively
- one output array c, size $m + n$

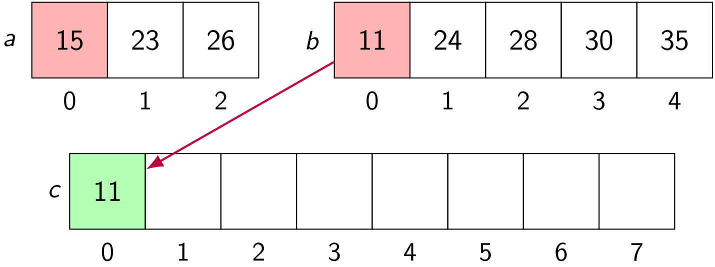
a	15	23	26
	0	1	2

b	11	24	28	30	35
	0	1	2	3	4

c								
	0	1	2	3	4	5	6	7

Merge 1

- compare, then copy smallest
- then increment b and c indices



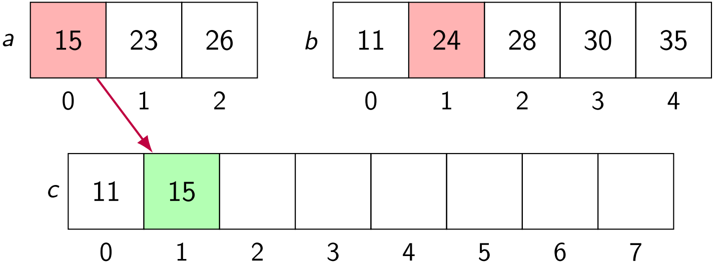
a	15	23	26
	0	1	2

b	11	24	28	30	35
	0	1	2	3	4

c	11							
	0	1	2	3	4	5	6	7

Merge 2

- compare, then copy smallest
- then increment a and c indices



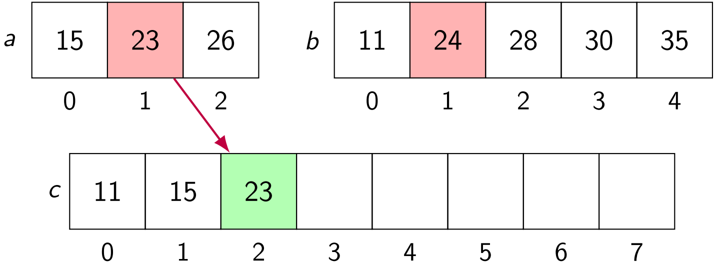
	0	1	2
a	15	23	26

	0	1	2	3	4
b	11	24	28	30	35

	0	1	2	3	4	5	6	7
c	11	15						

Merge 3

- compare, then copy smallest
- then increment a and c indices



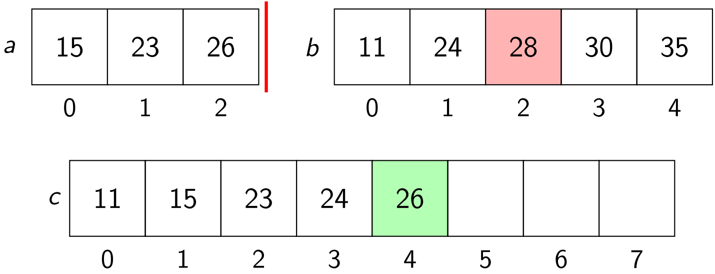
a	15	23	26
	0	1	2

b	11	24	28	30	35
	0	1	2	3	4

c	11	15	23					
	0	1	2	3	4	5	6	7

Merge 5

- eventually, the a or b index falls off its array
- what then?



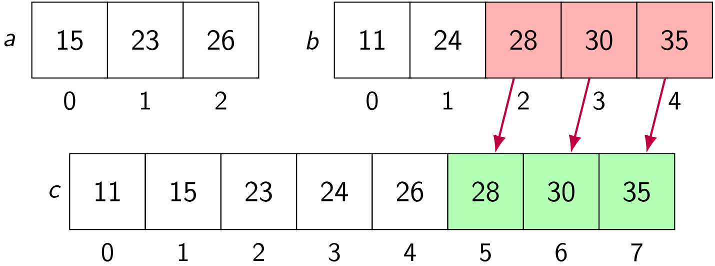
a	15	23	26
	0	1	2

b	11	24	28	30	35
	0	1	2	3	4

c	11	15	23	24	26			
	0	1	2	3	4	5	6	7

Merge Completion

- whichever of a or b has elements left
- simply copy remaining elements with no more comparisons



a	15	23	26
	0	1	2

b	11	24	28	30	35
	0	1	2	3	4

c	11	15	23	24	26	28	30	35
	0	1	2	3	4	5	6	7

Merge Analysis

- input size?
- end early?
- arrangement matter?
- analysis?

Merge Analysis

- input size? # elements of a and b: $n + m$
- end early? no
- arrangement matter? yes (there is a best and worst case)
- analysis?

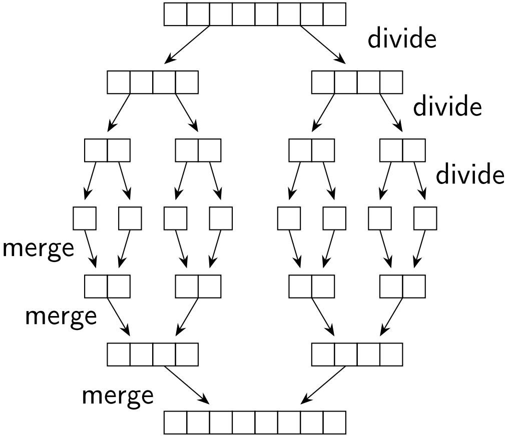
$$T(n) \in \Theta(n)$$

Merge Implementation

- a brief note on implementation
- in reality, there are not three arrays (a, b, and c)
- there is only one array, partitioned into two contiguous ranges
- the “output” is simply the combined range

- questions on merge?

Mergesort



Mergesort Analysis

consult code

- input size?
- operations?

- number of recursive calls?
- size of each recursive call?
- end early?

Mergesort Analysis

consult code

- input size? range from left to right
- operations? comparison, assignment, couple of arithmetics, and merge $\in \Theta(n)$
- number of recursive calls? 2
- size of each recursive call? $n/2$
- end early? no

recurrence relation?

Mergesort Analysis

consult code

- input size? range from left to right
- operations? comparison, assignment, couple of arithmetics, and merge $\in \Theta(n)$
- number of recursive calls? 2
- size of each recursive call? $n/2$
- end early? no

recurrence relation? $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$

- a?
- b?
- d?

Mergesort Analysis

consult code

- input size? range from left to right
- operations? comparison, assignment, couple of arithmetics, and merge $\in \Theta(n)$
- number of recursive calls? 2
- size of each recursive call? $n/2$
- end early? no

recurrence relation? $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$

- a? 2
- b? 2
- d? 1

- final analysis?

Mergesort Analysis

consult code

- input size? range from left to right
- operations? comparison, assignment, couple of arithmetics, and merge $\in \Theta(n)$
- number of recursive calls? 2
- size of each recursive call? $n/2$
- end early? no

recurrence relation? $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$

- a? 2
- b? 2
- d? 1

- final analysis? $T(n) \in \Theta(n \lg n)$