Scientific Computing

Mon, Feb 9

Announcements

* Homework 2 due this Friday, 11:59pm
  pdf + zip file on D2L

Don't forget to keep track of and cite any external resources you use - friends, websites, AI, etc.

Two kinds of things to cite:

(1) A resource helped me learn about a topic

(2) A resource wrote this line of code. Be specific.

* Also, written explanations should be your own words.

Office Hours:

Mon, 9:30-10:30
Fri, 2:00-3:00

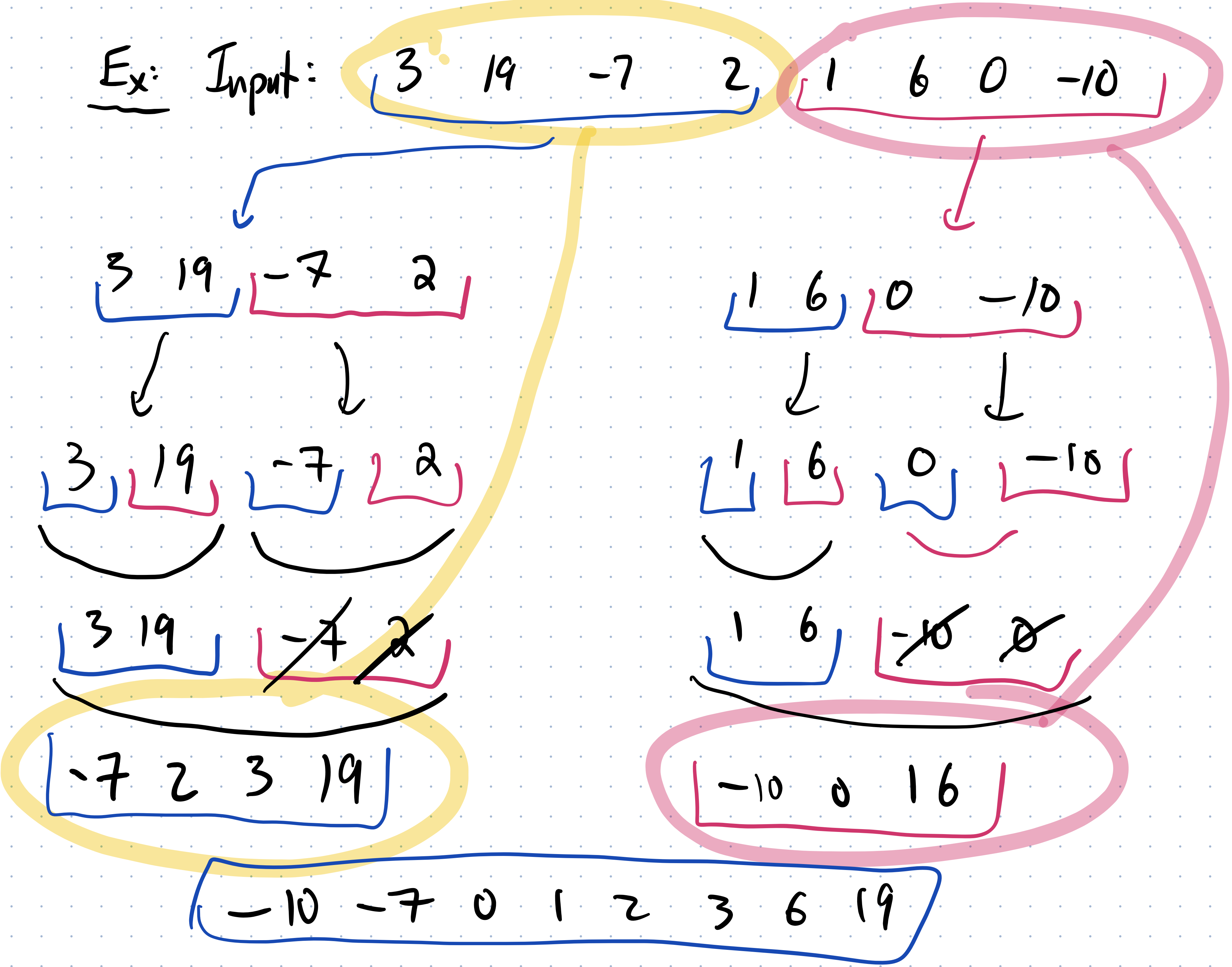
Cudahy 307

Topic 7 - Divide and Conquer

"Divide and Conquer" is an algorithmic paradigm that is roughly

1) Split the input in half
2) Solve the problem on each half separately (recursively)
3) Combine your two answers into one big answer.

Ex: Input:


3 19 -7 2 1 6 0 -10
3 19 -7 2
1 6 0 -10
3 19 -7 2
1 6 0 -10
3 19 -7 2
1 6 -10 0
-7 2 3 19
-10 0 1 6
-10 -7 0 1 2 3 6 19

Pseudocode

```
function merge_sort(Q):        Q = list of #s
  if |Q| ≤ 1:        ] If the input list has a single element, it's already properly sorted so return it
      return Q
  L = left half of Q
  R = right half of Q
  L = merge_sort(L)
  R = merge_sort(R)
               (at this point, we get to assume L and R are individually sorted)
  new_list = []
  while |L|+|R| > 0:
     take L[0] or R[0], whichever is smaller,
        remove it, and add to new_list
  return new_list
```

recombine

What's the runtime? Harder, because it's recursive. What we can do is find a recurrence for the runtime. $a_n = a_{n-1} + a_{n-2}$

Suppose the runtime is $T(n)$ when the input has size n.

Steps:

Apply to left half	$T(n/2)$
Apply to right half	$T(n/2)$
Merge	n

Recurrence: $T(n) = 2T(\frac{n}{2}) + n$

There is a theorem called The Master Theorem that tells you how to convert a recurrence into a formula.

see Wikipedia page

In this case, it tells us:

$T(n) = O(n \log(n))$.

⇝ Jupyter Notebook Sorting demo