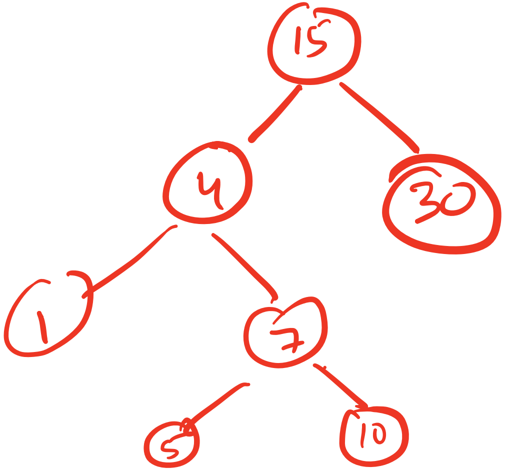
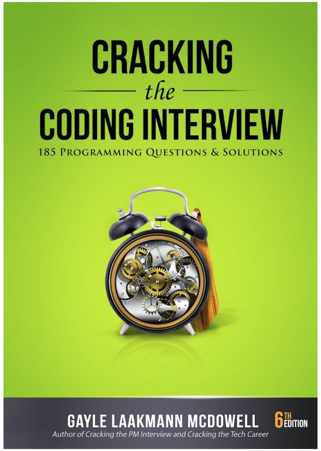
INTERVIEW PRACTICE

Tips for Technical Interviews (coding)

#1 Listen carefully, look for unique info

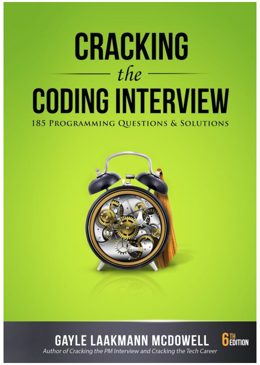
#2 Draw an example.




CRACKING
the
CODING INTERVIEW
185 Programming Questions & Solutions
GAYLE LAAKMANN MCDOWELL
6TH EDITION
Author of Cracking the PM Interview and Cracking the Tech Career

#64-67

Tips for Technical Interviews



1. Listen carefully
2. Draw an example
3. State the brute force or a partially correct solution
   - then work to get at a better solution
4. Optimize:
   - Make time-space tradeoffs to optimize runtime
   - Precompute information: Reorganize the data e.g. by sorting
5. Solidify your understanding of your algo before diving into writing code.
6. Start coding!

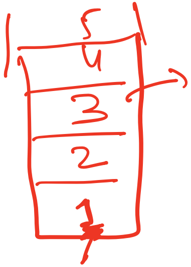
Small group exercise

Write a ADT called minStack of numbers that provides the following methods

O(1)
- push() // inserts an element to the “top” of the minStack
- pop() // removes the last element that was pushed on the stack
- top () // returns the last element that was pushed on the stack

~~O(1)~~ ?
- min() // returns the minimum value of the elements stored so far

STL Stack

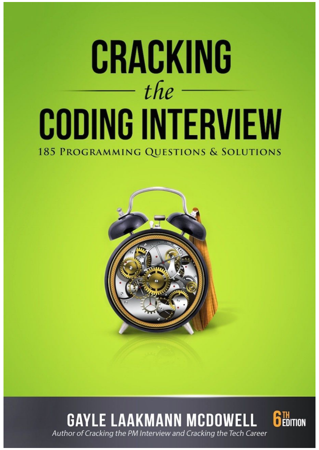


5 4 3 ② ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~7~~

6

min : ~~5~~ ~~4~~ ~~3~~ ~~2~~ 1

5 1 ① 4 3 ~~X~~ ~~X~~



CRACKING
the
CODING INTERVIEW
185 Programming Questions & Solutions
GAYLE LAAKMANN MCDOWELL
6TH EDITION
Author of Cracking the PM Interview and Cracking the Tech Career

1. stl stack + variable to keep track min

2. <u>Stack</u> + priority-queue] We discussed issues with this solution if the algo for push was to insert the key to both the stack & priority-queue

✓

push O(1) + O(log n) = O(log n)

pop O(1) + O(log n)

top O(1) stack

min <u>O(log n)</u> priority ~~queue~~ / min ~~heap~~

3. <u>stack</u> + <u>sorted</u> vector

✓

O(N)

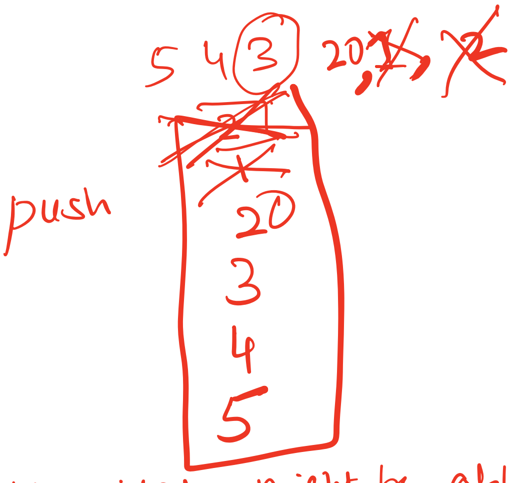
<u>push()</u> O(N) O(1) → stack + O(N) to remove from sorted

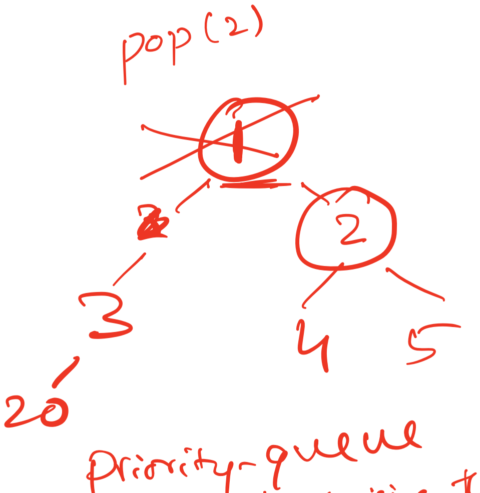
<u>pop()</u>

top() O(1) → stack

min O(1) → sorted vector

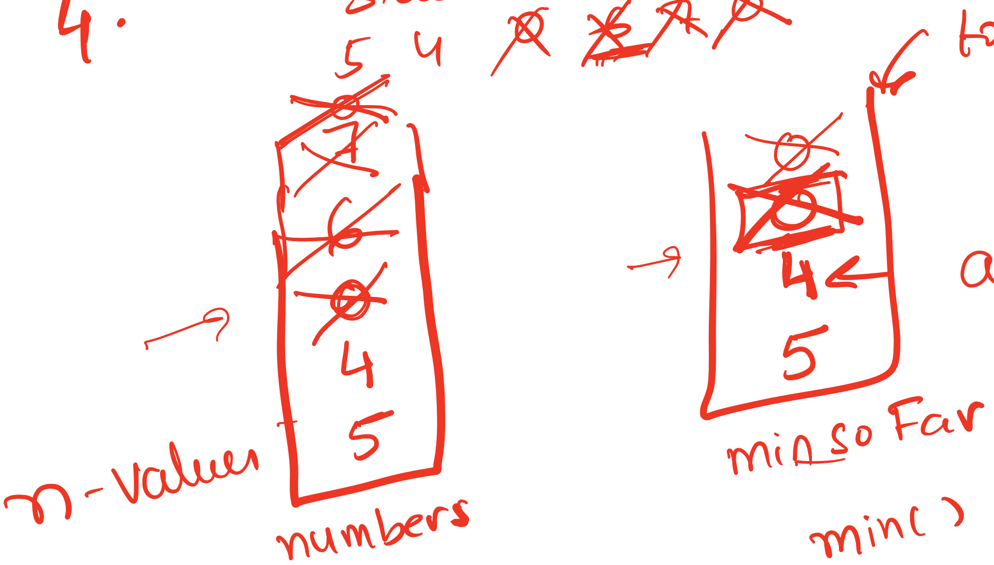
Example run showing the issue with using a stack and priority queue (with a specific algo in mind)





note: you might be able to make this work by revising the algo discussed in class for push

4. Stack + Stack or Stack of pairs



top : min()

atmost n value

O(n)+O(n)

min() O(1)

push pop top

O(1) push() push numbers stack

if val ≤ top of minsofar

push into minsofar

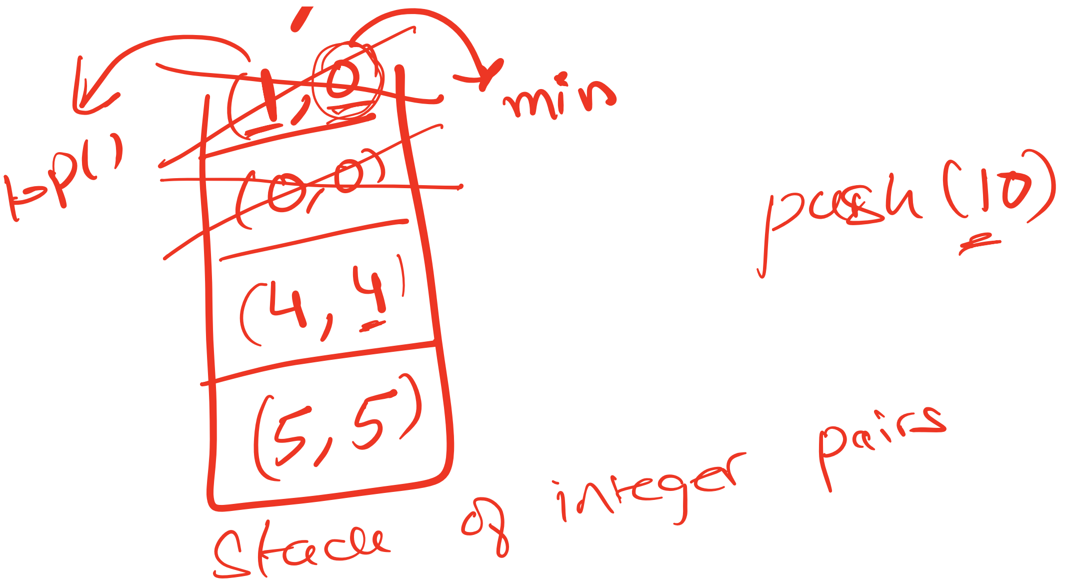
O(1) min()

O(1) pop() pop out of both stacks

O(1) top()

Space complexity O(n)

10.



5. Unsortedvector : (last element pushed to the vector is the top of the stack)

O(1) push: push to end of vector

O(1) pop : remove from end of vector

O(1) top : return last element in the vector

O(n) min : will need to examine all elements to find the min