SORTING
DATA STRUCTURE SELECTION
INTERVIEW PRACTICE

Sorting a forest of Binary Search Trees

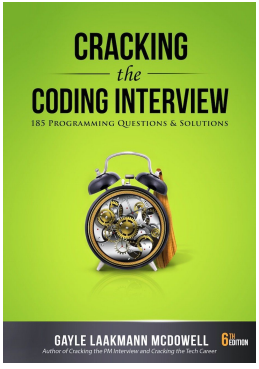
Refer to lecture code.

```
//Precondition: unsorted vector
//Post condition: sorted vector in ascending order
void selectionSort(vector<int>& a, int N){

   for(int i =0; i<N; i++){
      int index=i;
      for(int j = i+1; j<N;j++){
         if(a[j]<a[index]){
            index = j;
         }
      }
      int tmp = a[i];
      a[i] = a[index];
      a[index]=tmp;
   }
}
```

* Modify selection sort to work on any type of data
* Select a data structure that can help speed up the running time

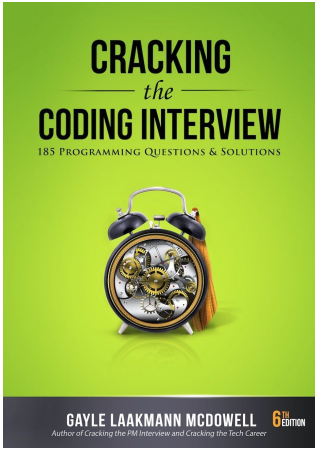
Technical Interviews



Some tips for interviews

1. Listen carefully
2. Draw an example
   - that is specific & large enough to be interesting
3. State the brute force or a partially correct solution (then debug 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!

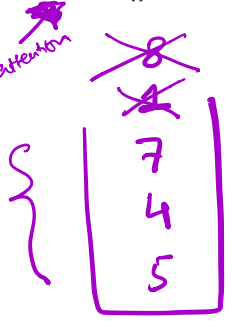
Small group exercise



Write a ADT called a minStack that provides the following methods

- 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
- min() // returns the minimum value of the elements stored so far

need! pay attention



push: 5, 4, 7, 1, 8

min: 5, 4, 4, 1, 1

top: 5 4 7 1 8

pop() // 8 deleted

min() // 1

top() // 1

pop() // 1 deleted

min() // 4

top() // 7

Questions

- what kind of elements does minStack store? int.
- Can I use C++ STL classes? yes
- what is the difference between top() and min()?