

SORTING DATA STRUCTURE SELECTION INTERVIEW PRACTICE

Sorting a forest of Binary Search Trees

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

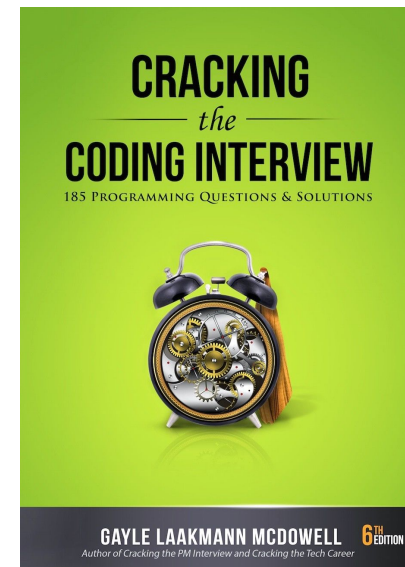
    for(int i =0; i<N; i++){
        int index=i;
        for(int j = i+1; j<N;j++){
            if(a[j]<a[index]){
                index = j;
            }
        }
        int tmp = a[i];
        a[i] = a[index];
        a[index]=tmp;
    }
}
```

- * 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 in 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

