3. Guidelines

Tuesday, February 1, 2022 2:57 AM

Checklist:

1) Good use of data structures
2) Avoid Code Re-use | Do not repeat yourself (DRY)
3) Less than $O(n^2)$. Avoid nested loops
   - 2 separate loops is better
4) Low space-complexity
   - Recursion can cause stack overflow, copying large arrays may exceed memory.

Guidelines (not always applicable):

1) Hashmaps is often used to improve time complexity
2) Binary Search Tree (BST) is great for sorted arrays w/ $O(\log N)$ time
   - Divide & conquer - divide a data set into smaller chunks & repeating a process w/ a subset of data
3) Try sorting inputs
4) Hashmaps and precomputed data (i.e. sorted arrays) can optimize time well
5) Time & space tradeoff. Sometimes storing extra state in memory can improve runtime
   - Creating Hashmaps uses more space, but can greatly optimize time