

Top-20 Training Program (DataStructure Design Problems)

Apply the solution building strategies discussed in class to solve following problems.

Group1: Min Stack & Queue

MinStack: <https://leetcode.com/problems/min-stack/description/>

MinQueue: Do the same as previous problem but on queue datastructure.

Group 2: Random DS

Random DS-I: <https://leetcode.com/problems/insert-delete-getrandom-o1/description/>

Random DS-II: <https://leetcode.com/problems/insert-delete-getrandom-o1-duplicates-allowed/description/>

All-O(1)-DS: <https://leetcode.com/problems/all-oone-data-structure/description/>

Group 3: Cache variations

LRU Cache: <https://leetcode.com/problems/lru-cache/description/>

LFU Cache: <https://leetcode.com/problems/lfu-cache/description/>

Bidirectional Symbol Table: Create a data structure that supports following operations in $O(1)$ time:

- put(key, value)
- getByKey(key)
- getValueByValue(value)

This kind of datastructure is very useful in DNS lookup and DNS reverse lookup i.e., domain name to IP resolution and viceversa.

Inverted index of web: Given a list of web pages, create an inverted index for the words of those web pages i.e., associate with each word a list of web pages in which that word appears. Write a program that reads in a list of web pages, creates an inverted index, and support multi-word queries. That means, it must output the list of web pages that contain at least one occurrence of each of the query words.