Interview Notes

Michael Li

Contents

Interview notes I have. Dicusses various types of problems, approaches to these problems, and example Leetcode problems

1 Prefix Sum

Technique calculates the sum of elements in an array using **prefix sums**, which are the sum of the first k elements of an array

$$p_0 = 0 \qquad p_1 = a_0 \qquad p_k = a_0 + a_1 + \cdots + a_k$$

The prefix sums can be calculated in $O(n)$ using the following formula

$$p_i = p_{i-1} + a_i$$

A similar idea can be applied for **suffix sums**, which are the sum of the last k elements of an array

Prefix sums allow for quick calculation of the sum of any **slice** of the array. For example, suppose we want to calculate the sum of m slices $[x..y]$ of an array of length n.

Normal bruteforce solution takes $O(m \cdot n)$ runtime since we need to iterate through the array to calculate the sum of each slice

If we instead calculated the prefix sums of the array, call this prefix sum array P, we could find the sum of any slice from

$$P[y+1] - P[x]$$

This changes the runtime to $O(n + m)$

- n from calculating the prefix sums
- m from doing an $O(1)$ operation m times for finding the sum of m slices using prefix sum array P

NOTE the indices, we want to include all elements at index y and below, but none of the elements below index x

1.1 Prefix Sum Code

Calculate the prefix sums given an array nums of length >= 1

```
public int[] prefixSums(int[] nums) {
  int[] prefixSums = new int[nums.length];
  prefixSums[0] = nums[0];
  for (int i = 1; i < prefixSums.length; i++) {
    prefixSums[i] = prefixSums[i-1] + nums[i];
  }
  return prefixSums;
}
```

1.2 Leetcode Problems

- 1480. Running Sum of 1d Array
- 724. Find Pivot Index