Prefix array

Given an array with length , and queries of the form . For each query, output .

If directly calculate, each query , then total time complexity !

Important concept: precomputation

Precompute useful stuff first, then use it to answer queries.

prefix array:

how to answer ?

!!!

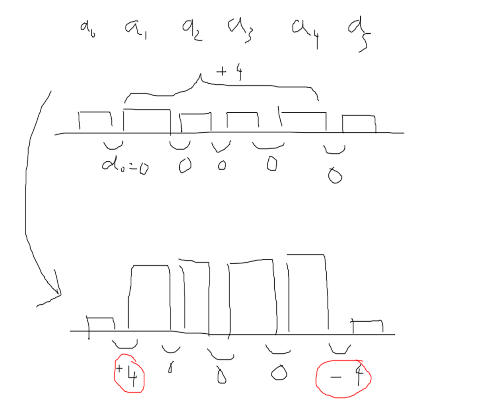
wow! precomputation, per query, total time !

Advanced: this concept can be extended, for example some problem may ask number of integers from to that satisfy some condition. Then, we can calculate number of integers from to and number of integers from to !

also there’s a function called partial\_sum in <numeric>

Difference array

Given an array of length , and updates of the form . For each update, add to all elements to . Output the final array after the updates.



Again, if naïve, per update, total time complexity .

difference array:

update :

a useful transformation: consider the difference instead of the elements!

two pointers

very general technique, a bit hard to say which situations to use and how to use

Maintain 2 pointers in an array, traverse the array with 2 pointers at the same time

sometime take advantage of a sorted array

Most of the case one of the pointer will be in a for loop

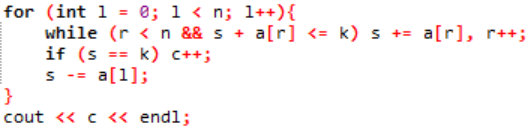
2 types: equal direction, opposite direction

a bit hard to describe so ill just give a bunch of examples

Example 1: find number of subarrays with sum equal to

(subarray: continuous chunk)

brute force:

Maintain and , such that

!

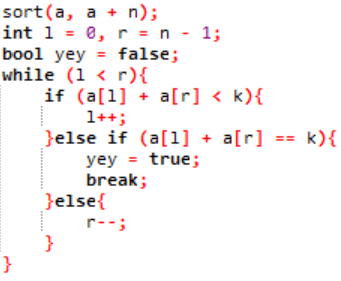
Example 2: find if any is equal to

brute force:

Sort the array, initialize as and as

if : move to the right once

if : move to the left once



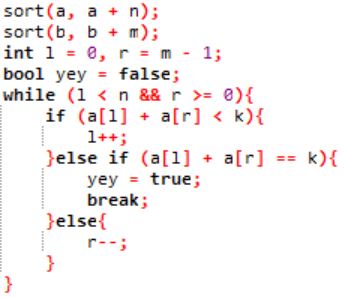
Example 3: find if any is equal to

pretty much same as above, just to show that you can apply 2 pointers in different arrays

actually similar to merging arrays in merge sort

sort both arrays, ,

if : move to the right once

if : move to the left once

Example 4: find if a triplet sums to

3 pointers? nahh

notice is smaller, can

just iterate the , then use 2 pointers like above to find if any !

too lazy to code

Example 5: find if a quadruplet sums to

4 pointers??? no

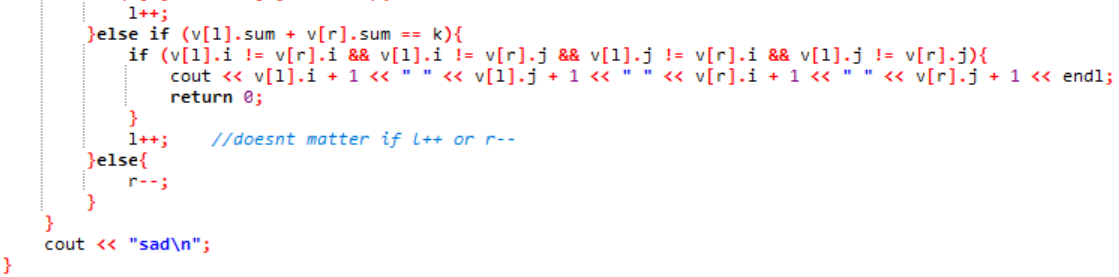
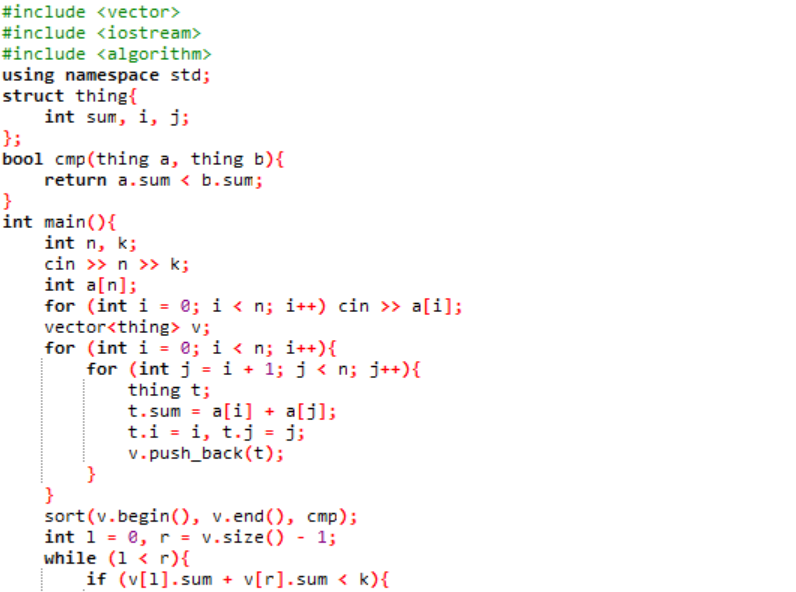
can be decomposed into 2 pointers + 2 pointers!

First, find the sum of all possible pairs, and store them in an array.

Use example 2 to do 2 pointers on the array with sums!

Important: need 2 check if the pairs does not have overlapping elements.

time complexity = =

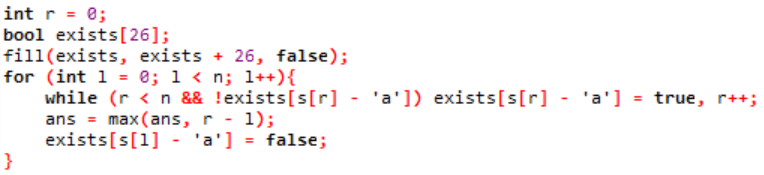


type of 2 pointers: sliding window

maintain and , and is in a for loop. There is a property for a subarray, and you need to maintain such that to is the longest subarray that satisfies the property

Like example 1!

Example 6: Find the length of the longest substring without repeating characters

maintain such that to is the longest substring without repeating characters

Final notes

prefix array and difference array are useful for thinking about problems from a different perspective, and it is also easy to understand.

2 pointers is a bit hard to get, so just do more 2 pointers problems!

2 pointers problems: S061, M1613, M1812, M1022, M0652, I0611, a lot more

advanced topics: difference array on trees, Floyd’s tortoise and hare