

# Leetcode Virtual Contest - 282

Problem & Difficulty - Easy

Stat: Counting words with a given prefix.

Cond<sup>n</sup>: given an array of strings words and a string pref return the number of strings in words that contains pref as a prefix.

Input: words = ["pay", "attention", "practice", "attend"], pref = "at"  
Output: 2

## Thought Process —

Prefix = ??

the word which add  
in starting of given  
word.

Here, after read problem statement  
I get an idea of `string::find()`  
fun.

but this fun find in  
whole string of our  
target word.

we only want  
at zero index,  
for that ↓

`(if (found) = String::npos ++ found)`



$found \neq string$  and  $found == 0$   
  
 the word not present in string.  
 found only at zero index.

Code:    // take cnt variable     $int\ c = 0;$   
           // iterate in string     $for(auto\ s: 'words')$   
           //  $auto\ found = s.find(pref);$   
           // if  $(found \neq string::npos \ \&\ found == 0) \rightarrow c++$   
           // return c;

## Problem-2 {Difficulty - Medium}

Stat: Minimum number of steps to make two strings  
 Anagram II

Condition: given string  $s$  and  $t$ . In one step, you can append any character to either  $s$  or  $t$ .

Return min<sup>m</sup> number of step to make  $s$  and  $t$  anagram of each other.

Input:  $s = "leetcode"$ ,  $t = "couts"$

Output: 7



Anagram = ??

both string contain same type  
of characters and same no.  
of times.

Thought Process —

first thing after reading problem  
statement that come on mind  
i.e

→ find frequency of character in  
both string. So which data structure  
provides this → ??

↓  
unordered-map → is used.

String1 = "right"  
↓ store in map.

String2 = "thing"

|            |                |        |
|------------|----------------|--------|
| $m[r] = 1$ | $\swarrow = 0$ | $m[t]$ |
| $m[i] = 1$ | $\swarrow = 0$ | $m[h]$ |
| $m[g] = 1$ | $\swarrow = 0$ | $m[i]$ |
| $m[h] = 1$ | $\swarrow = 0$ | $m[n]$ |
| $m[t] = 1$ | $\swarrow = 0$ | $m[g]$ |



if char not found stores as  $[-1]$  otherwise

decrease count if  $found$

eg ② String ① = "leetcode"

String ② = "roats"

$m[l] = 1$

$m[t] = 2$

$m[a] = 1$

$m[c] = 1$

$m[o] = 1$

$m[d] = 1$

$m[e] = 1$

$m[c]$

$m[o]$

$m[a] = -1$

$m[t]$

$m[s] = -1$

take abs  
value of  
both

$1+1=2$

output:  $1+2+1+1+2$   
 $= 7$

code: // unordered\_map<char, int> m;  
// for (auto c: s)  
     $m[c]++$ ;       $\rightarrow found$   
// for (auto c: t)  
     $m[c]--$ ;       $\rightarrow not\ found$

// int cnt = 0  
for (auto it: m)  
     $cnt += abs(it.second)$   
return cnt;



### Problem-③ {Difficulty- Medium}

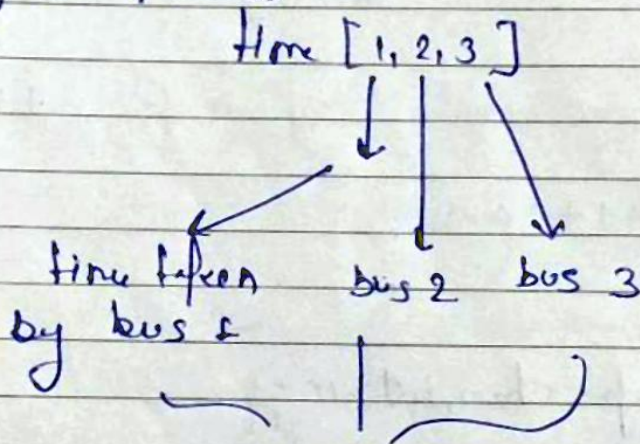
Stat: Minimum time to complete trips.

Cond<sup>n</sup>: given an array time where  $time[i]$  denote the time taken by the  $i$ th bus to complete the trip.

Input:  $time = [1, 2, 3]$ ,  $totaltrips = 5$

Output: 3

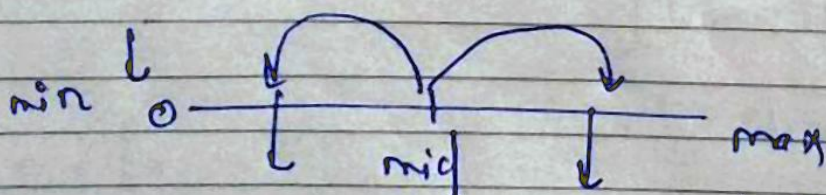
Thought Process —



they can run simultaneously.

Step-① We set time for bus = 0  $\rightarrow$  min and then check the condition.

take one max time = INT\_MAX.



we can complete trips less than given.

we can take more time to complete trips.



life —

$$\min = 0$$
$$\max = 10^{14}$$

{ given in constraint }

$$10^7 \times 10^7$$

↓      ↓

time    heap

$$\text{mid} = \frac{0 + 10^{14}}{2}$$

now we take  $\max = 100$

$$\text{mid} = \frac{0 + 100}{2} = 50$$

|   |    |   |
|---|----|---|
| ↓ | 12 | 3 |
|---|----|---|

↓      ↓      ↓

50    50/2    50/3

      = 25    = 16

data structure  
= Binary search

$$50 + 25 + 16 > 5$$

$$\text{now check} = \frac{0 + 50}{2} = 25$$

$$(25 + 12 + 8) > 5$$

$$\text{now again} = \frac{0 + 25}{2} = 12$$

$$(12 + 6 + 4) > 5$$

$$\text{now } \frac{0 + 12}{2} = 6$$

$$(6 + 3 + 2) > 5$$

$$\text{now} = \frac{0 + 6}{2} = 3$$

$$(3 + 1 + 1) = (8) \rightarrow \text{here}$$

at time = (3).    total heap = (8)



Code: low=0, high=1000000000000000, ans=high

// while (low < high)

mid = (low + high) / 2

cnt = 0

for (i = 0 to km)

cnt += (mid / time[i])

// if (cnt >= total trips)

ans = min(ans, mid), high = mid

// else

low = mid + 1

// return ans;