



# PEPCODING

PURSUIT OF EXCELLENCE AND PEACE

MODULE

## HASHMAP & HEAP



+91 11 4019 4461



PepCoding, 3rd Floor, 15 Vaishali,  
Pitampura Opposite Metro Pillar 347,  
Above Karur Vysya Bank, New Delhi,  
Delhi 110034



[www.pepcoding.com](http://www.pepcoding.com)



/pepcoding

## Valid-Invalid Approach

Q Smallest Window in a string Containing all the characters of another string.

What?

what!  
Find the length of the smallest window that contains all the characters of other string.

## How?

- We use Valid-Invalid approach
  - Make Frequency map of second string.

eq:

$$g: a \text{ b} \rightarrow a d e d d b c c d a a b a c d a c$$

$$P = abbcad$$

pMap for p =  $a \rightarrow 2, b \rightarrow 2, c \rightarrow 1, d \rightarrow 1$   
 $= a^2b^2c^1d^1$

st  
—10

a<sub>0</sub> b<sub>1</sub> c<sub>2</sub> a<sub>3</sub> d<sub>4</sub> e<sub>5</sub> d<sub>6</sub> d<sub>7</sub> b<sub>8</sub> c<sub>9</sub> c<sub>10</sub> d<sub>11</sub> a<sub>12</sub> a<sub>13</sub> b<sub>14</sub> a<sub>15</sub> c<sub>16</sub> d<sub>17</sub> a<sub>18</sub> c<sub>19</sub>  
 ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑  
 end (mention P  
back so  
what's)

$SMap \Rightarrow a \rightarrow X2$

$b \rightarrow Y_2$

C → 1

$$d \rightarrow x \neq 3$$

match count = 123456.

when match count = characters

of P.

then start moving st.

When character gained by end add that character to hash map. If current frequency of character (before updation) is less than frequency of that char in PMap then increment match count.

Match count ~~also~~ keeps no. of characters matching b/w SEP.

Position at which start is the character **H** is recently removed from word document

(II) a<sub>1</sub>, b<sub>2</sub>, c<sub>3</sub>, a<sub>4</sub>, d<sub>5</sub>, a<sub>6</sub>, b<sub>7</sub>, c<sub>8</sub>, d<sub>9</sub>, a<sub>10</sub>, a<sub>11</sub>, b<sub>12</sub>, a<sub>13</sub>, c<sub>14</sub>, d<sub>15</sub>, a<sub>16</sub>, c<sub>17</sub>, d<sub>18</sub>, a<sub>19</sub>  
 ↓ ↓  
 st  
 end.

map a-2 b-2 c-1 d3  
at end = 18) 1       $m_c = \$ 5$   
Since we crossed a character of use  
Smpfrncr(char)  $\leftarrow$  pmap(char)

Since it has changed we will argue in windows.

since it has changed we will acquire in windows.

$$ws \rightarrow window\ st = we = (st+1) = (-1+1) = 0 \quad [when \\ we=window\ end = \boxed{we = end = 8} \quad . \quad \begin{array}{l} \text{matchout} \\ \text{is same} \\ \text{place} \end{array}]$$

map:  $a_1 b_2 c_3 d_4$        $mc = 56$   
 $\frac{1}{2} \quad \frac{2}{3}$       at(12)

After start a2 b2 c3 d4 mc = 5

$st=1$  end=12  $(12-1+1) > (8-0+1)$  so no update.

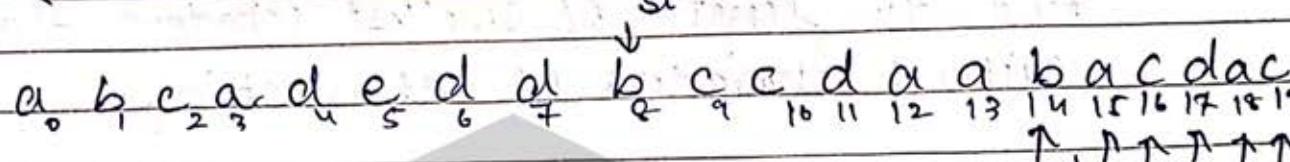
(III) 

$$\text{map} = \begin{smallmatrix} 2 & 1 & 0 & 1 & 0 & 1 \\ 3 & 2 \end{smallmatrix} \quad m c = 5/6$$

$$\text{Start} = \begin{matrix} a & b & c & d \\ \text{m} & 2 & 1 & 5 \end{matrix} \quad mc = 45$$

when start was at 8 end at 14  $\text{gap} = (14 - 8 + 1)$   
 $= 7$   
 $\text{gap} < (8 - 0 + 1)$

b [ws = 8, we = 12].

(iv) 

smap =  $\begin{matrix} a & \cancel{b} & c & \cancel{d} & \cancel{e} \\ 3 & 4 & 5 & 6 & 7 \end{matrix}$  mc = 5

match count remain 5 so no update.

Ans ws = 8 we = 14

String : bccdaab.

(\*) Acquire till matchcount is not equal to characters in p. i.e acquire till valid.

(s) settle  $\rightarrow$  calculate the size of window on the basis of current start and end. It better update.

(r) Release  $\rightarrow$  Release character till match count remains same. as soon as matchcount decreases settle.

Q Smallest window containing all characters of yourself (smallest distinct windows)

what?

Find the length of smallest window that contains distinct characters of this string.

How?

Same approach as before.  
Make hashset of the string.

A musical staff with six measures. The first five measures have vertical stems pointing down. The sixth measure has a vertical stem pointing up, followed by a vertical bar line, and then two vertical stems pointing down.

(I) map =  $\begin{matrix} ax & b1 & c1 & d1 \\ x \\ \hline 3 \end{matrix}$

hashset

$\langle a, b, c, d \rangle$ .

(map size = set size)

So found a window.

$$st = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\text{gap} = \begin{matrix} 0 & 1 & 5 & 6 \end{matrix}$$

(II) map = a2 c1 d1/x3 b1  
st = a2 c1 d3 b1

~~11100~~ 15 (4)

— 1 —

(III) map = a2 d3 b1 c1  
st = a2 d3 b1 c1

Any 4 sides.

(IV) map  $\rightarrow$   $d \frac{x}{y} b y_2 c y_2 a$

dbac

$$st = \frac{dy}{dz} \frac{bx}{z^2} \frac{c_2}{z} \alpha_1$$

(V) map = d1 b1 a1 c1  
st = d'1 b1 a'1 c1

## Q) longest substring with unique characters.

What?

Find the length of the longest substring which has unique character i.e. no character is repeating.

How?

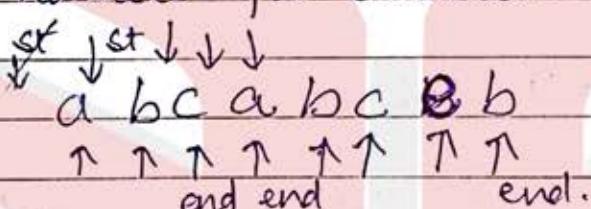
Valid Invalid approach.

a\* → acquire till no duplicacy.

s → compare the length of current string with maxlen.

r → release a character.

Make Hashset for characters.

eg: 

(I) set =  $\{a, b, c\}$  curLen = 3 maxlen = 3

After release  $\{b, c\}$

(II) set  $\{a, b, c\}$  maxlen = 3

release  $\{a, b\}$

(III) acquire  $\{a, c, b\}$  maxlen = 3

release  $\{b, c, e\}$

Ans = 4

III acquire  $\{a, c, b\}$   
 release  $\{a, b\}$

## Q Count substring with unique characters

What?

Count all the substrings with unique characters.

How?

Same as before Just instead of updating max length update count by  $(\text{end} - \text{st} - 1)$

eg:  $\begin{matrix} a & b & c & a & b & c & e & b \\ \uparrow & \uparrow \\ \text{st} & & & 3 & & & 6 & \text{end} \end{matrix}$

(I) Set  $\langle ab \rangle$

After release of  $a_0$ .

Set  $\langle b, c \rangle$

st = -1

end = 3

Settle count  $+ = (\text{end} - \text{st} - 1)$   
 $= (3 - (-1) - 1)$

(stopped aquiring with count = 3)

Release  $a_0$ . Set =  $\langle b, c \rangle$

(II) After aquiring set  $\langle b \& a \rangle$

end = 1

st = 0

Settle count  $+ = (4 - 0) = 3 + 3 = 6$

After release  $\langle c \& a \rangle$

(III) After aquiring  $\langle c \& a \& b \rangle$

Settle

count =  $6 + (5 - 1 - 1) = 9$

a

end = 5

st = 1

After release  $\langle a, b \rangle$

Similarly count = 22

Ans = 22

## Q Longest Substring with K Unique Characters

What?

Count find the longest substring that has K unique characters.

How?

Same approach as all unique characters were valid till unique character count remains K.  
 Use hashmap to store characters of current subarray

acquire\* → acquire till count of unique characters is less than or equal to K. (add to map)

settle → calculate current length and compare with max length and update if greater.

release\* → release till count of unique characters is not less than K. (remove/decrease count from map)

eg:  $\begin{matrix} \downarrow & \downarrow \\ a_1, b_2, c_3, a_4, d_5, b_6, b_7, c_8, d_9, d_{10}, a_{11}, c_{12} \\ \uparrow & \uparrow \\ \text{end.} & \text{end.} & & & & & & & & & & \text{end.} \end{matrix}$

(I) acquire\* till end = 5

$$\underline{\text{settle count}} = \text{end} - \text{st} - 1 = (5 - 1) - 1 = 5 \quad \text{max} = 5$$

release\* till st = 1 length remains

(II) acquire\* till end = 6

$$\underline{\text{settle}} = \text{end} - \text{st} - 1 = (6 - 1 - 1) = 4 \quad \text{max} = 5$$

release\* till st = 3

(III) acquire\* till end = 8

$$\underline{\text{settle}} = \text{end} - \text{st} - 1 = (8 - 3 - 1) = 4 \quad \text{max} = 5$$

release\* till st = 4

(IV)	acquire* till end = 11 settle count = $(11 - 4 - 1) = 6$ release* till st = 7	<u>max = 6</u>
(V)	acquire* till end = 13 settle count = $(13 - 1 - 7) = 5$ release* till st = 13	max = 6.

Ans = 6

Q Count Substring with K unique characters

What?

Count all the substrings that have K unique characters.

How?

We will use 2 hashmap to store characters of current subarray.

Small map → This map will acquire characters till size is  $< K$ .

Big map → This map will acquire character when valid ie no. of unique characters is  $\leq K$ .

Maintain 2 ends end1 & end2 one for each hmap.

acquire\* → If no. of unique characters in small map is less than K add current character to both small map & big map and increment both j1 & j2.

If  $(j_1 < j_2)$  then only acquire in small as characters already in big map.

~~count~~

When no. of unique characters in small is equal to K i.e size of small == K  $\Rightarrow$  big.size == K acquire in big till it has K unique characters. and increment j. end 2

Settle: Increment count by  $(\text{end2} - \text{end1} + 1)$

remove / release : Release the character from both maps . and increment count . till not valid to , acquire again

Eg:  $\begin{matrix} & \downarrow \\ a_0 & b_1 & b_2 & c_3 & b_4 & c_5 & c_6 & a_7 & a_8 & d_{11} & b_{10} & c_{11} \\ \uparrow & \uparrow \\ e_1 & e_2 & e_1 & e_2 \end{matrix}$

(I) Acquire\* - acquire in small till 3. end 1 = 8  
acquire in big till 9 end 2 = 9  
(this no.

$$\text{Count} = 9 - 4 + 1 = 6$$

(II) Acquire\* → acquire iff small as  $\text{end1} < \text{end2}$ .

small tend end 1 = 8 < b3 c3 a2 >

big till 9 end2=9 < b3 c3 a2 >

$$\text{count} = 6 + (9 - 8 + 1) = 8.$$

release = till st=1 count += 8 + (9-8+1) = 10

$$st2 = 2 \quad count += 1 \otimes + (2) = 12$$

$$st = 3 \quad \text{count} \pm = 10 \pm (5) = 11$$

st = 3 count += 1 + (2) = 14

$St=4$  again valid. (count is incremented as can't acquire)

(III) Acquire\* Acquire in small till  $\text{end1} = 10$   
 $\text{end2} = 10$

$$\text{count} = 14 + (10 - 10 + 1) = \underline{\underline{15}}$$

Release and count =  $\text{st} = 5$   $\text{count} = 15 + 1 = 16$   
 $\text{st} = 6$  (Valid again)

(IV) Acquire\*  $\rightarrow$   $\text{end1} = 11$   
 $\text{end2} = 11$

$$\text{count} = 16 + (11 - 11 + 1) = 17$$

Release and count  $\text{st} = 7 \Rightarrow 17 + 1 = \underline{\underline{18}}$

(V) Acquire\*  $\text{end1} = 12$   
 $\text{end2} = 12$

$$\text{count} = 18 + (12 - 12 + 1) = \underline{\underline{19}}$$

Ans 19

## & Equivalent Subarrays

What?

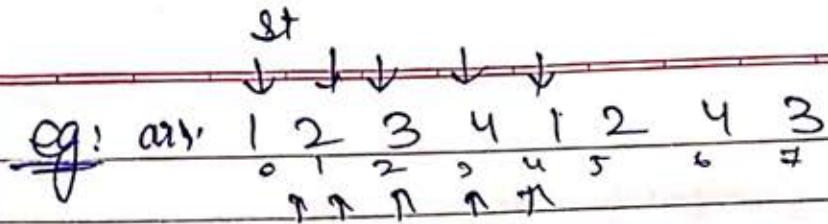
Count all subarrays containing all distinct character of given array. (Make Hashset of array)

How?

acquire\*  $\rightarrow$  acquire in map till our map.size is less than set.size.

count  $\rightarrow$   $\text{count} += (\text{array length} - \text{end} + 1)$  when  
 $\text{map.size} == \text{set.size}$

release  $\rightarrow$  Release and check if  $\text{set.size} == \text{map.size}$   
 count increment similar way as above.



$\text{Set} = \{1, 2, 3, 4\}$

(I) acquire till end = 3

$$\text{count} = (8 - 3) = 5$$

release st = 0

(II) acquire till end = 4

$$\text{count} + = (8 - 4) = 9$$

release st = 1

(III) acquire till end = 5

$$\text{count} + = (8 - 5) = 12$$

release st = 2

(IV) acquire till end = 7

$$\text{count} + = (8 - 7) = 13$$

release st = 3

count + = (8 - 7) = 14  
 (as map size = set.size)

Ans = 14

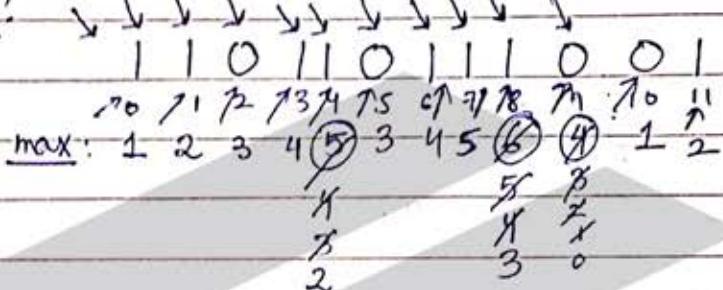
Q.

### Maximum Consecutive Ones (Ans)

- a) find maximum length substring which has maximum consecutive ones where substring can have at most one 0.

How?

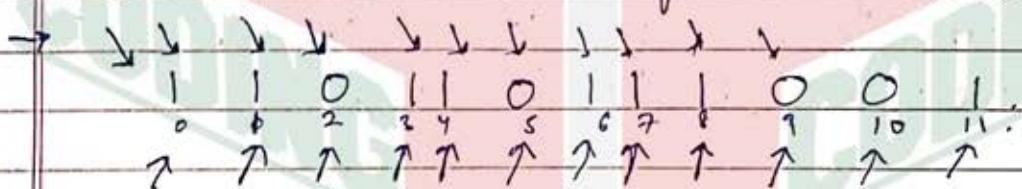
ex:



At each index  $i$ ,

- a) acquire till you are valid, i.e., till you have only 1 zero in your substring ( $\alpha^*$ )
- b) settle, i.e., update max. ( $\delta$ )
- c) release till you loose a zero. ( $\gamma^*$ )

→ Initially start = -1, end = 0 which means end will now acquire 0th index, and start will now leave first 0th index.



0 - 4 : end++ and it acquires, length = 5.

Now end at 5; update max = 5.

Since at 5 value is 0, start ++ till you release 0.

Now start at 2, length = 3.

Now end acquires till 8, and is at 9.

Update max = 6 and start leaves till it gets a 0 at 5.

Now end acquires 9th index, start leaves till 9th, and this works till you reach end of array.

b) This question has two more variations, one where k zeroes are allowed and k is given, other where k=0, i.e., only 1's substring.

ex:  $k = 2$ .

$\begin{matrix} > & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ & \diagup & \diagdown \\ \text{ans} = 6 & (0-5 / 4-9) \end{matrix}$

ex:  $k = 0$

$\begin{matrix} > & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ & \diagup & \diagdown & \diagup & \diagdown & \diagup & \diagdown & \diagup & \diagdown \\ \text{ans} = 4 & (0-3) \end{matrix}$

Q. Binary strings with substrings representing 1 to N (Ans).

what?

You are given a binary string, return true if the string can represent all numbers from 1 to N both inclusive. N will be given.

How?

For a binary string to represent 1 to N, if it represents  $\frac{N+1}{2}$  to N, then it represents all 1 to N.

This is because:  $1 \rightarrow 1$  if  $5 \rightarrow 101$   
 $2 \rightarrow 10$  if  $6 \rightarrow 110$   
 $3 \rightarrow 11$  if  $7 \rightarrow 111$   
 $4 \rightarrow 100$  if  $8 \rightarrow 1000$

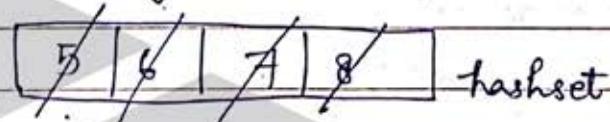
If 8 is present, 4 is also present. (if N is present,  $\frac{N}{2}$  is also present)

Use start-end approach, keep start = -1 and end = 0 initially.

Create a hashset of values  $N+1$  to  $N$ .

- acquire till you are invalid, i.e., binary substring represents a number  $\leq N$  ( $a^*$ ), if greater,
- release till the substring is valid again. stop.

ex:  $N = 8$



I)  $\begin{array}{ccccccc} & \downarrow & & & & & \\ 0 & 1 & 0 & 1 & 0 & ] & 1 1 0 1 1 1 0 0 0 0 1 0 \\ \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & & \\ \text{number} & 0 & 1 & 2 & 5 & \boxed{10} & \end{array}$

$10 > 8$ , release till valid.  
 [Cut 5 in hashset]

II)  $\begin{array}{ccccccc} & \downarrow & & & & & \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 ] 0 1 1 1 0 0 0 0 1 0 \\ \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow & \\ \text{number} & 0 & 1 & 2 & 5 & \boxed{11} & \end{array}$

$11 > 8$ , release.

III)  $\begin{array}{ccccccc} & \downarrow & & & & & \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 0 ] 1 1 0 0 0 0 1 0 \\ \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow & \\ \text{number} & 0 & 1 & 2 & 3 & 6 & \boxed{13} \end{array}$

$13 > 8$ , release

IV)  $\begin{array}{ccccccc} & \downarrow & & & & & \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 0 ] 1 0 0 0 0 1 0 \\ \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow & \\ \text{number} & 0 & 1 & 2 & 5 & \boxed{11} & \end{array}$

Cut 6 in hashset

$11 > 8$ , release.

V)  $\begin{array}{ccccccc} & \downarrow & & & & & \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 0 ] 0 1 1 1 0 \\ \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow & \\ \text{number} & 0 & 1 & 3 & 7 & \boxed{13} & \end{array}$

$13 > 8$ , release.

VI)  $\begin{array}{ccccccc} & \downarrow & & & & & \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 0 ] 1 0 0 0 1 0 \\ \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow & \\ \text{number} & 0 & 1 & 3 & 7 & \boxed{12} & \end{array}$

Cut 7 in hashset

(2 release shown here)

$12 > 8$ , release.

VII)  $\begin{array}{ccccccc} & \downarrow & & & & & \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 0 ] 1 0 0 0 1 0 \\ \uparrow & \nearrow & \nearrow & \nearrow & \nearrow & \nearrow & \\ \text{number} & 0 & 1 & 2 & 4 & 8 & \end{array}$

Cut 8 in hashset,  
 return true.

Q. Find subarray with given sum. ( $A^n S$ )

what?

Find a subarray with sum to a target.

How?

- # Use start - end approach,
- acquire till your sum is  $\leq$  target.
- if your sum == N, settle, i.e. return answer.
- release till your sum is  $<$  target.

ex:  $\rightarrow [2, 8, 3, 4, 9, 5]$       start = -1 &  
 initially  
 end = 0  
 target = 16

I) start = -1      end =  $\emptyset / \neq 3$   
 array =  $[2, 8, 3, 4]$   
 sum =  $\cancel{2} 10 \cancel{13} 17 \rightarrow$  invalid, so release now.

II) start = -1 & 0      end = 3.  
 array =  $\cancel{[2, 8, 3, 4]}$ .  
 sum =  $\cancel{17} 15 \rightarrow$  valid, so acquire again.

III) start = 0      end =  $\cancel{3} 4$   
 array =  $\cancel{[2, 8, 3, 4, 9]}$   
 sum =  $\cancel{15} 24 \rightarrow$  invalid, release now.

IV) start =  $\emptyset / 1$       end = 4.  
 array =  $\cancel{[2, 8, 3, 4, 9]}$   
 sum =  $\cancel{24} 16 \rightarrow$  sum == N.

return  $[3, 4, 9]$  as answer.

// Similar question:

Q. Find subarray with product less than K.  
 → Just calculate product instead of sum.

## Anagrams Based

Q. find all anagrams  
what?

You are given a string and a pattern, return count of all anagrams of the pattern in the string.

How?

# Use Match Count strategy to do this (MC) where match count tells how many characters are being matched in current string and pattern string.

→  $a^*$  : with each acquire, if frequency of element in current map is less than that in pattern map, then MC++, else if it is equal or greater, MC does not change.

→  $s$  : If MC == pattern.length(), count of anagram is increased by 1.

→  $r^*$  : Release a character after checking that if its frequency in current map was less or equal to that in pattern map, then MC--, else MC does not change.

ex: string: badcc dabb adaccab cd b

pattern: abbcd

pattern map:  $\begin{array}{|c|c|c|c|} \hline a & \rightarrow 1 & b & \rightarrow 2 \\ \hline c & \rightarrow 1 & d & \rightarrow 1 \\ \hline \end{array}$   
fmap

1)  $\begin{array}{cccccccccccccccc} b & a & d & c & c & d & a & b & b & a & d & a & c & c & a & b & c & d & b \\ \hline 1 & 0 & 1 & 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \end{array}$

Current map:

$\begin{array}{|c|c|c|c|} \hline b & \rightarrow 1 & a & \rightarrow 1 \\ \hline d & \rightarrow 1 & c & \rightarrow 2 \\ \hline \end{array}$

$b_0 \downarrow q \downarrow d_2 \downarrow c_3 \downarrow$  when  $b_0$  was released.

MC = ① ② ③ ④ ⑤ When  $c_4$  is acquired, current map size. == fmap size, so release as MC not = 5.

2)  $\overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{b}}$

MC 1 2 3 4 5  
 current map  $a \rightarrow 1 | c \rightarrow 2 | d \rightarrow 2$   
 $MC = 2$

3)  $\overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{b}}$

$a \rightarrow 1 | c \rightarrow 2 | d \rightarrow 2 | 1$   
 $MC = 2/3$   
 release d.  $\Rightarrow MC = 3$ .

4)  $\overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{b}}$

$a \rightarrow 1 | b \rightarrow 1 | c \rightarrow 2 | d \rightarrow 1$   
 $MC = 3/4$   
 Release c  $\Rightarrow MC = 4$ .

5) acquire

5)  $\overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{b}}$

$a \rightarrow 1 | b \rightarrow 2 | c \rightarrow 1 | d \rightarrow 1$   $\Rightarrow MC = 4/5$  count = 1.

6)  $\overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{b}} \overrightarrow{\text{a}} \overrightarrow{\text{d}} \overrightarrow{\text{a}} \overrightarrow{\text{c}} \overrightarrow{\text{c}} \overrightarrow{\text{a}} \overrightarrow{\text{b}} \overrightarrow{\text{c}} \overrightarrow{\text{d}} \overrightarrow{\text{b}}$

$a \rightarrow 1$	$b \rightarrow 2$	$c \rightarrow 1$	$d \rightarrow 1$
$a \rightarrow 2$	$b \rightarrow 1$	$c \rightarrow \emptyset$	$d \rightarrow 1$
$\emptyset$	$1$	$2$	$1$
$2$	$1$	$2$	$1$

$MC = 5/4$  (release c<sub>4</sub>)  
 $\beta$  (release d<sub>5</sub>),  $\gamma$  (acquired<sub>10</sub>)  
 $\beta$  (r b<sub>7</sub>),  $\gamma$  (a c<sub>12</sub>),  $\beta$  (a b<sub>8</sub>)  
 $\beta$  (ac d<sub>10</sub>),  $\beta$  (a b<sub>15</sub>)  
 $\gamma$  (a d<sub>17</sub>),  $\beta$  (a b<sub>18</sub>)  
 $\rightarrow$  count = 2.

∴ We have 2 anagrams of abcd,  
 which are cdabb and abcdb  
 4 5 6 7 8                    14 15 16 17 18

Q. K-anagrams.  
 What?

Tell if 2 strings are K-anagrams of each other.

Now?

ex: s1: dba abcd a                    k=3.  
 s2: abc d abc d.

1) Make fmap of s1.

d → 2	b → 2	a → 3	c → 1
-------	-------	-------	-------

2) Update fmap using s2 such that if element exists in fmap, frequency --.

After update: 

d → 0	b → 0	a → 1	c → -1
-------	-------	-------	--------

# frequencies > 0 denote extra characters in s1, and < 0 denote extra characters in s2.

3) Sum either all frequencies > 0 or < 0 and if sum  $\leq k$ , return true.

Here, sum = 1 where  $1 \leq 3$ , ∴ these strings are K-anagrams.

Q. Are two arrays equal?  
 What?

Return, if 2 strings are same / 2 arrays are same.  
How? K anagrams where  $k=0$ .

## Q. Anagram Mappings what?

You are given 2 strings which are anagrams of each other, return an array which tells the indexes of a character in string 2 present in string 1. Randomizing is done as mapping where  $P[i] = j$  where character at  $i$  in string 1 is at  $j$  in string 2.

How? # Return i indexes corresponding to each j array.

1. Without duplicacy: Make hashmap  $\langle \text{Char}, \text{Int} \rangle$  where each character stores index of it in string 1.  
→ Now create an array by iterating in string 2, for each character, find it in hashmap and add its index in array.

ex:  $s1: f c d h a e b g$   
      ◦ 1 2 3 4 5 6 7

$s2: c a f d h b g e$   
      ◦ 1 2 3 4 5 6 7.

HashMap: 
$$\begin{array}{l} f \rightarrow 0 \\ d \rightarrow 2 \\ a \rightarrow 4 \\ b \rightarrow 6 \\ c \rightarrow 1 \\ h \rightarrow 3 \\ e \rightarrow 5 \\ g \rightarrow 7 \end{array}$$

$s2: c a f d h b g e$   
Iterate ans: 
$$\left[ \frac{1}{1} \ 4 \ \frac{0}{2} \ \frac{2}{3} \ \frac{3}{4} \ \frac{6}{5} \ \frac{7}{6} \ \frac{5}{7} \right]$$

2. With duplicacy: Make hashmap of character vs. either arrayList to store multiple indexes and always add in last in list to avoid remove first operation, or use a priority queue / a linked list.

ex: s1: f d b a c e d g f a c g b h  
 0 1 2 3 4 5 6 7 8 9 10 11 12 13

s2: a b b g e f d f g h c c a d  
 0 1 2 3 4 5 6 7 8 9 10 11 12 13

HMAP: ~~f → 8, 0~~

char vs Priority Queue (ascending order).

f → 0, 8

d → 1, 6

b → 2, 12

a → 3, 9

c → 4, 10

e → 5

g → 7, 11

h → 13

Now iterate:

s2: a b b g e f d f g h c c a d  
 0 1 2 3 4 5 6 7 8 9 10 11 12 13

ans: [ 3 2 12 7 5 0 1 8 11 13 4 10 9 1 6 ]  
 ↓  
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 ]

# if you use arraylist, and do add last while adding indexes & remove last while removing indexes, then just make the ans array from right to left ( $\leftarrow$ )

### Q. Valid Anagram What?

You are given 2 strings S and T, return 'yes' if T is an anagram of S or 'no' if it is not.

How?

# Simply create a fmap of S, and if any character of T is not in fmap, or frequency of any character in S is > 0 after cancelling with character frequencies in T, return false.

ex: S: pepcoding  
T: dongpcpie

Ip → /e → /l c → /l o → /l d → /l i → /l n → /l g → /l  
! ! ! ! ! ! ! ! ! ! ! !

returns true as all o and all characters present.

### Q. Group Anagrams

What?

Given an array of strings, return a 2D list which has each list as a group of anagrams. Case-sensitive anagrams are not grouped together, like Abc and abc.

How?

We keep a HashMap of HashMap vs ArrayList.

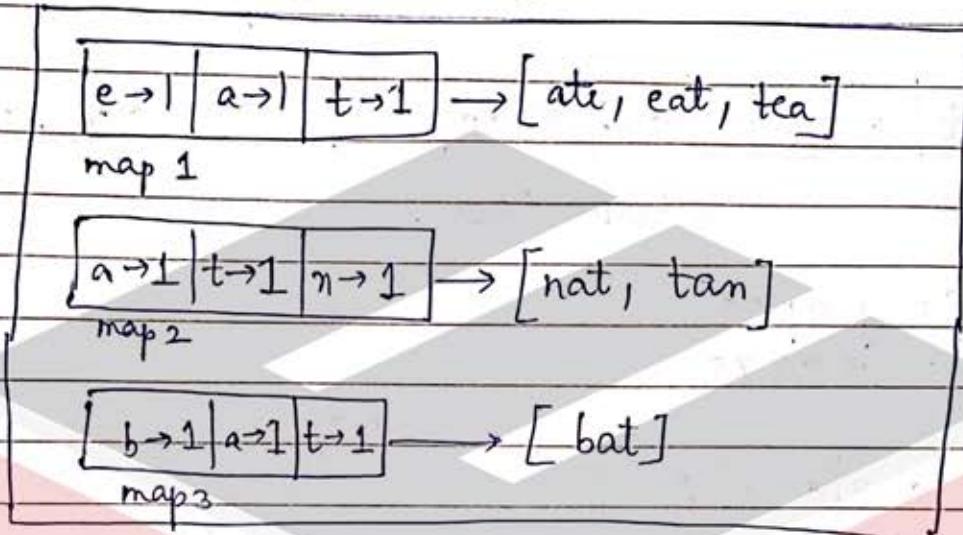
HashMap<Char, Int> which will be frequency map for each string.

for each fmap, it will have all words having the same fmap.

ex: [cat, tea, tan, ate, nat, bat].

$\text{HM} < \text{HM} < \text{Char, Int} \rangle, \text{ArrayList} \rangle$  map.

map



ex // Make fmap for ate, check if fmap already exists in map, if yes, add to its list ate, else create a new entry in map for fmap vs list having ate.

### Q Isomorphic strings what?

Given two strings S and T, tell if they are isomorphic. Isomorphic strings are those in which we can map the characters of one string to characters in other string.

How? Create two maps, and at any given index i,  
 $\text{map1}[S.\text{charAt}(i)] = T.\text{charAt}[i]$  and  
 $\text{map2}[T.\text{charAt}(i)] = S.\text{charAt}[i]$ .

If it already exists and the character to be mapped with is different than what is stored, return false.

ex1 S: paper  
 T: title.

map1

$p \rightarrow t$	$a \rightarrow i$	$e \rightarrow l$	$r \rightarrow e$
-------------------	-------------------	-------------------	-------------------

map 2

$t \rightarrow p$	$i \rightarrow a$	$l \rightarrow e$	$e \rightarrow r$
-------------------	-------------------	-------------------	-------------------

returns true.

ex2 S: paper  
 T: ~~aaaaaa~~.

map1

$p \rightarrow x$	$a \rightarrow x$	$e \rightarrow x$	$r \rightarrow x$
-------------------	-------------------	-------------------	-------------------

map 2

$x \rightarrow p$
-------------------

? fails.

// this example explains why checking & creating both maps is important.

## Q. Word Pattern

What?

You will be given pattern string and words string, check if words follow exact order as pattern string.

How? # Same as isomorphic strings above, just characters are mapped to strings in map1, and strings to characters in map2.

ex: S: abab      W: pep code pep code.

map1

$a \rightarrow \text{pep}$	$b \rightarrow \text{code}$
----------------------------	-----------------------------

map2

$\text{pep} \rightarrow a$	$\text{code} \rightarrow b$
----------------------------	-----------------------------

returns true.

## Prefix Sum Based Questions

Q) subarray Range with given target (count subarray with sum k)  
What?

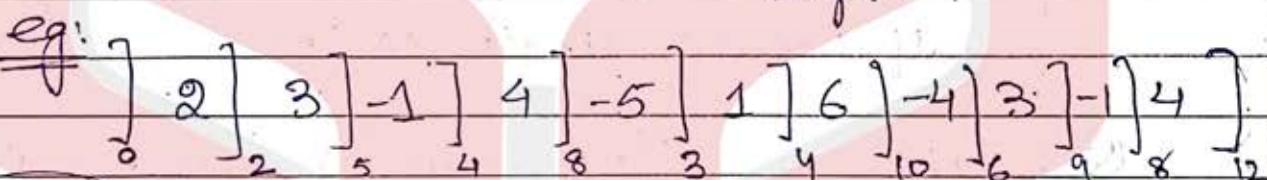
Count all the subarrays of a array that has sum equal to given target.

How?

→ Make a hashmap that stores sum (prefixsum) as key and count of occurrence of that sum as value.

FMAP → PS v Count

target = 4



Initially added

$0 \rightarrow 1$  10-1

$2 \rightarrow 1$  6-1 Total count

$5 \rightarrow 1$  9-1

$4 \rightarrow 1$  12-1

$8 \rightarrow 1$

$3 \rightarrow 1$

$\downarrow$   
 $4 \rightarrow 0$  present  
 so count + 1

$= \cancel{0} \cancel{1} \cancel{2} \cancel{3} \cancel{4} \cancel{5} \cancel{6} \cancel{7} \cancel{9}$

Ans = 9

At each sum check if  $(\text{sum} - \text{target})$  is present or not. If present then increase Total count with frequency of  $(\text{sum} - \text{target})$ .

\* Increase Count ~~then after~~ before frequency update.

Q subarray with zero sum

what?

count all the subarrays whose sum is equal to zero.

How?

\* same strategy as before here target = 0

eg:	[	0	2	3	-2	-1	-2	4	-2	-2	]	0
				5	3	2	0	4	2	0		

$$0 \rightarrow 2 \quad 5-1 \quad 4-1 \\ 2 \rightarrow 2 \quad 3 \rightarrow 1$$

$$\text{count} = 1 \times 2 \times 5 \times 7 =$$

check if same prefix sum is in map or not  
if present update count.

Q subarray with sum divisible by k

what?

count all subarrays whose sum is divisible by k.

How?

same strategy as before with map.

here map will store (prefix sum % k.) as key and value will be count of this remainder.

eg:  $\downarrow$   $K = 5$

$$\begin{array}{ccccccccc} 2 & 3 & 1 & 4 & -18 & 6 & 3 & 4 & -14 & 2 & 9 \\ \boxed{2} & \boxed{3} & \boxed{1} & \boxed{4} & \boxed{-18} & \boxed{6} & \boxed{3} & \boxed{4} & \boxed{-14} & \boxed{2} & \boxed{9} \\ 0 & 2 & 5 & 6 & 17 & -1 & 5 & 8 & 12 & 0 & 9 \end{array}$$

$0 \rightarrow \cancel{X} \cancel{X} \cancel{X} 4$

$1 \rightarrow \cancel{X} 1$

$2 \rightarrow X X 3$

$3 \rightarrow X X 3$

$4 \rightarrow X 2$

Count =  $\cancel{X} \cancel{X} \cancel{X} 5 / \cancel{X} \cancel{X} 13$   
 $= \underline{\underline{13}}$

(here when  $Ps = -1$ )

$$\begin{aligned} \text{key} &= (-1 \% 5) + 5 \\ &= -1 + 5 \\ &= 4 \end{aligned}$$

\* In case of negative sum

$$\text{key} = (\text{prefixsum \% } K) + K$$

\* At each  $ps$  (remainder) check if already present or not. If present then increment count by frequency.

\* when no. are repeating it means all the numbers in between has sum divisible by  $K$

Q

Count subarrays with ~~sum~~ equal 0's & 1's

what?

Count all such subarrays in a array of 0's & 1's where 0's & 1's are equal.

how?

since zero & 1's should be equal thus we can do this by replacing all 0's with -1's and apply 0.sum, array strategy

When 0 & 1 are equal then only sum will be zero.

eg:  $\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$   
 $\begin{array}{ccccccc} 0 & 1 & 0 & 1 & 0 & 1 & 2 \end{array}$   $\downarrow$  1 < Prefix sum.

$0 \rightarrow \cancel{X} 2$

considered  
-1

Count =  $\cancel{1} \cancel{2} \cancel{4} \cancel{6}$   
 $\text{at}(idx_2) \text{ at}(idx_3) \text{ at}(idx_4)$

$1 \rightarrow \cancel{X} \cancel{2} 3$

$2 \rightarrow 1$

Ans = 9

9  
(idx = 6)

## Q Binary Subarray with Sum.

Same Question as subarray with sum equal to target here just array is binary. Use same strategy.

target = ~~9~~ 9

eg:  $\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$   
 $\begin{array}{cccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array}$

$0 \rightarrow \cancel{X} 2$

$3 \rightarrow \cancel{1}$

$6 \rightarrow \cancel{1}$

count = ~~0~~ ~~1~~ ~~4~~ ~~5~~ ~~7~~ ~~9~~

$1 \rightarrow \cancel{X}$

$4 \rightarrow \cancel{X} 2$

$7 \rightarrow \cancel{X} 2$

$10 \rightarrow \cancel{1} 1$

$2 \rightarrow \cancel{X} 2$

$5 \rightarrow \cancel{X} 2$

$11 \rightarrow \cancel{1} 1$

Ans = 16

## Q Count Substrings with equal no. of 0's, 1's & 2's

What?

Count the no. of substrings with equal no. of zeros & ones & twos.

How?

- Make Hashmap where key is a string made like  $(\text{count}_1 - \text{count}_0) * + (\text{count}_2 - \text{count}_1) * + \dots$  any character

Eg:

	1	0	2	1	0	0	2	1	1
count0	0	0	1	1	1	2	3	3	3
count1	0	1	1	1	2	2	2	3	4
count2	0	0	1	1	1	1	2	2	2
0*0	1*-1	0*-1	0*0	0*-1	0+1	-1*-1	-1*0	0*-1	-1*2

HMAP

$$0*0 = Y 2$$

$$1*-1 = Y 2$$

$$0*-1 = Y 3$$

$$-1*-1 = 1$$

$$-1*0 = 1$$

$$-1*2 = 1$$

$(2-1) * (1-2)$

$$2*-1$$

$$\text{count} = X 2 \quad 3 \quad 5$$

at(2) at(1) at(0) at(1)

$$\text{Ans} = \underline{5}$$

When count was 2 of 0\*-1

it means 2 of 0\*-1

can make combination

with third 0\*-1

Why?

When delta of counts (string) is repeated again it means that no. of increments in 0's is same to 1's and no. of increments in 1's is same to 2's. So this results in substring with equal no. of zero's & 1's

Q) Longest substring with equal no. of 0's & 1's.

What?

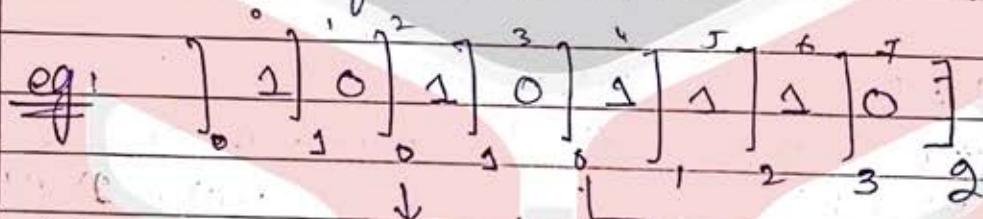
Find the longest substring which has equal no. of 0's & 1's.

How?

Using same strategy as count substring / subarray with equal no. of zeroes & 1's.

Consider 0 as -1.

Here we will check the length b/w first occurrence of sum and current occurrence.



Map

0 → -1

1 → 0

2 → 5

3 → 6

sum vs index

$$\text{gap} = 1 - (-1) = 2$$

$$\max \text{len} = \cancel{2} \checkmark 4 \text{ at } (1) \text{ at } (3)$$

$$\text{gap} = 3 - (-1) = 4$$

at index 4

$$\text{gap} = 4 - 0 = 4$$

some updation

Ans = 4

Q) Longest Subarray with sum equal to k.

What?

Find the longest subarray whose sum is equal to K.



How?

Same strategy as sum equal to target. Just save the first occurrence of prefix sum and calculate gap instead of count.

target = 4

eg:	$\left[ \begin{matrix} 2 \\ 3 \\ -1 \\ 4 \\ -5 \\ 1 \\ 6 \\ -4 \\ 3 \\ -1 \\ 4 \end{matrix} \right]$	$\left[ \begin{matrix} 0 \\ 2 \\ 5 \\ 4 \\ 8 \\ 3 \\ 9 \\ 10 \\ 6 \\ 9 \\ 8 \\ 12 \end{matrix} \right]$	$\downarrow$	gap = 4-4	gap = 6	gap = 7	gap = 6	gap = 7	gap = 7	max len
$0 \rightarrow 1$	$8-3$									
$2 \rightarrow 0$	$3-4$	$(ps-K)$		gap = $i - map.get(ps-K)$						= 3
$5 \rightarrow 1$	$10-6$	present			$= 2 - (-1) = 3$					<del>6</del> 7
$4 \rightarrow 2$	$6-7$									
$9 \rightarrow 8$	$12-10$									
										<u>Ans = 7</u>

MAP PS vs index.

Q longest subarray with sum divisible by K

Same strategy as above. Just store sum as prefix sum modulus K. And for negative sum (prefix sum  $\cdot K$ ) + K. Calculate the Gap.

K=5

eg:	$\left[ \begin{matrix} 2 \\ 3 \\ -1 \\ 7 \\ 4 \\ -18 \\ 6 \\ 3 \\ 4 \\ -14 \\ 2 \\ 9 \end{matrix} \right]$	$\left[ \begin{matrix} 0 \\ 2 \\ 5 \\ 6 \\ 13 \\ 17 \\ 5 \\ 8 \\ 12 \\ 10 \\ 11 \\ 19 \end{matrix} \right]$	$\downarrow$	gap = 2	gap = 4	gap = 5	gap = 4	gap = 8	gap = 11	
ps vs index										
$0 \rightarrow 1$										
$2 \rightarrow 0$										
$1 \rightarrow 2$										
$3 \rightarrow 3$										
$4 \rightarrow 5$										
										<u>Ans = 11</u>
										$gap = [i - map.get(ps)]$

## Q) Largest Subarray with zero sum

Same as largest subarray with sum equal to target here sum is zero (target=0). solve similarly.

## Q) Smallest Subarray with all occurrences of most frequent Element

what?

Find the smallest subarray of the array which also has  $x$  as the maximum frequency element.

How?

Make a pair class that stores 3 properties for each element.

Pair → frequency (frequency of element)  
 ↳ start (start of subarray)  
 ↳ end (end of subarray)

Initially calculate the frequency and make a pair for each element and store in hashmap.

e.g.: 4° 2' 3° 1° 2' 1° 4° 2' 4° 3° 1°

element vs Pair ob (freq, st, end)

4 → (0, 0, 0)

bfr bf bs bend.

2 → (1, 1, 7)

1 1 0 0

3 → (1, 2, 2)

2 2 1 4

1 → (1, 3, 1)

1 2 3 4 5

5 → (1, 3, 1)

2 3 1 7

gap small  
so - change

Ans = [2 3 1 2 1 4 2]

On first occurrence frequency is set to 1 and start index of element. Add this pair to hashmap.

When frequency increases check for bf if current frequency greater than bf. Update best frequency value (bfv), best start (bs), best end (bend) (best frequency)

When frequency same as bf check if current subarray is smaller than best subarray. If yes then update (gap = end - st + 1). The bf, bfv, bs, bend.

Subarray bw bs & bend is smallest subarray.  
 $\text{length} = (\text{bend} - \text{bs} + 1) = (7 - 1 + 1) = \underline{\underline{7}}$

Q Count of substrings containing k ones.

What?

Count all the substrings in a binary array that contains k ones.

How?

Solve in same strategy as count subarray with sum equal to target. Here target will be k.

target = k (Just array is binary else same)

## HashMap Questions

Q. Rat in a Jungle  
what?

In a forest, each rat has some colour. Some subset of rats tell you how many other rats have same colour as them.

Return minimum number of rabbits that could be in a forest.

How?

ex: [ 2, 2, 2, 2, 4, 4, 4, 6, 6, 6, 3, 3, 3, 3, 3, 3, 7, 7 ]

Create a fmap where  $\text{map}[a[i]] += 1$ .

i.e; fmap:	$2 \rightarrow 4$	<u>✓</u> <u>✓</u> <u>✓</u> <u>✓</u> <u>—</u> <u>—</u> = 6
	$4 \rightarrow 3$	<u>✓</u> <u>✓</u> <u>—</u> <u>—</u> = 5
	$6 \rightarrow 3$	<u>✓</u> <u>✓</u> <u>—</u> <u>—</u> = 7
	$3 \rightarrow 6$	<u>✓</u> <u>✓</u> <u>✓</u> <u>✓</u> <u>✓</u> <u>✓</u> = 8
	$7 \rightarrow 2$	<u>✓</u> <u>—</u> <u>—</u> <u>—</u> <u>—</u> = 8
		<u>—</u> = 34 rabbit

Why?

At a given point, fmap  $\langle \text{key}, \text{value} \rangle$  said says that: value number of ~~people~~ <sup>rabbit</sup> said that there were key number of ~~people~~ <sup>rabbit</sup> more like me.

ex:  $2 \rightarrow 4$  : 4 rabbits said that there are 2 more rabbits like them.

→ If  $\text{key} \geq \text{value}$ , ans  $+ = \text{key} + 1$ .  
 else ans  $+ = (\text{key} + 1)$ . where nor is nor

number of rabbits calculated as  $\text{ceil}(\frac{\text{value} * 1.0}{\text{key} + 1})$ .

ex:  $2 < 4$ , ans  $+ = [\text{ceil}(\frac{4}{3}) * (\text{key} + 1)] = 2 * 3$ ,  $= \underline{\underline{6}}$ .

## Q. Double Pair Array.

what?

Given an array of integers with even length, return true iff it is possible to reorder it such that  $A[2*i + 1] = 2 * A[2*i]$  for every  $0 \leq i < \text{len}(A)/2$ .

How? // Arrange such that every odd index is its previous even index element's twice.

ex:  $A = [-96, -96, -72, -48, -48, -48, -36, -24, -24, -12, 0, 0, 2, 2, 3, 3, 4, 4, 4, 6, 6, 6, 8, 12]$   
 to be considered

1) For each of these, first we create a fmap :

fmap

$-96 \rightarrow 2$	$-72 \rightarrow 1$	$-48 \rightarrow 3$	$-36 \rightarrow 1$	$-24 \rightarrow 2$
$-12 \rightarrow 1$	$0 \rightarrow 2$	$2 \rightarrow 2$	$3 \rightarrow 2$	$4 \rightarrow 3$
$8 \rightarrow 1$	$12 \rightarrow 1$			

2) Sort the array A .

3) Now, for each element, these things are considered:

a) If  $\text{fmap}[A[i]] == 0$ , continue.

b) If  $A[i] == 0$  and  $\text{fmap}[A[i]] \% 2 != 0$ ,  
 return false.

→ Because, 0 can occur only even times for  
 creating a double pair array.

c) If  $\text{fmap}.\text{containsKey}(A[i] * 2)$  and  $\text{fmap}[A[i] * 2] < \text{fmap}[A[i]]$ , return false.

→ Because, if we have  $2, 2, 2, 4, 4, \dots$ ,  
 then  $2, 4, 2, 4, \dots$  is possible. Third 2 has  
 no more ( $2^2$ ) i.e. 4 element left.

d) If above all are checked, then simply make  
 $\text{fmap}[A[i]] -= \text{minv}$  and  $\text{fmap}[A[i] * 2] -= \text{minv}$   
 where  $\text{minv} = \min(\text{fmap}[A[i]], \text{fmap}[A[i] * 2])$ ;

ex solved: for sorted array:

a) -96 :  $A[i] = -96$ ,  $fmap[A[i]] = 2$   
 $A[i]^*2 = -48$ ,  $fmap[A[i]^*2] = 3$   
 $\Rightarrow [-96, \underset{1}{-48}, \underset{2}{-96}, \underset{3}{-48}]$

$$fmap[A[i]] = 0$$

$$fmap[A[i]^*2] = 1$$

b) -72 :  $[-72, \underset{4}{-36}]$

c) -48 :  $[-48, \underset{6}{-24}]$

d) -24 :  $[-24, \underset{8}{-12}]$

e) 0 :  $[0, 0]$

f) 2 :  $[\underset{12}{2}, \underset{13}{4}, \underset{14}{2}, \underset{15}{4}]$

g) 3 :  $[\underset{16}{3}, \underset{17}{6}, \underset{18}{3}, \underset{19}{6}]$

h) 4 :  $[\underset{20}{4}, \underset{21}{8}]$

i) 6 :  $[\underset{22}{6}, \underset{23}{12}]$

∴ The array is:  $[-96, -48, -96, -48, -72, -36, -48, -24,$   
 $-24, -12, 0, 0, 2, 4, 2, 4, 3, 6, 3, 6,$   
 $4, 8, 6, 12]$

\* Also check that if there is a negative odd number, then return false [as no double pair array can be formed using -55.]

### Q. Is Frequency Equal what?

Given a string with lowercase alphabets, check if its possible to remove at most one character from this string in such a way that frequency of each distinct character is same in string.

below?

Consider 3 cases:

- a) a2 b2 c2 d2 : when all have same frequency.
- b) a2 b3 c2 d2 : when 1 has extra character.
- c) a3 b3 c3 d1 : when 1 character is extra.

ex: a x x y y z z.

fmap: x: 3  
 y: 2  
 z: 2.

\* Calculate max and min frequency and elements with  $\max_f$  &  $\min_f$  as  $\max_c$  &  $\min_c$ .

Here,  $\max_f = 3$ ,  $\min_f = 2$ ,  $\min_c = 2$ ,  $\max_c = 1$ .

→ If ( $\min_f == \max_f$ )  
 return true

→ If ( $\max_c == 1$  &  $\min_c == \text{fmap.size}() - 1$   
 &  $\max_f - \min_f == 1$ ) // Answer.  
 return true

→ If ( $\max_c == \text{fmap.size}() - 1$  &  $\min_c == 1$  &  
 $\min_f == 1$ )  
 return true.

→ else  
 return false.

### Q. Check AP Sequence

What?

Given an array of N integers, check whether AP can be formed using all these elements?

How?

ex: 0, 12, 4, 8.

1) Make a hashset

{ 0 | 12 | 4 | 8 }

2) Calculate 'd' of AP as.  $d = (\max - \min) / (\text{array size} - 1)$

$$\Rightarrow d = (12 - 0) / 3 \\ = 4$$

3) Now check if  $\min + d * i$  exists in set or not.

$$\min = 0, d = 4.$$

$$1) \min + d * 0 = 0 + 0 = 0. \checkmark$$

$$2) 0 + 4 * 1 = 4 \checkmark$$

$$3) 0 + 4 * 2 = 8 \checkmark$$

$$4) 0 + 4 * 3 = 12 \checkmark$$

All elements of hashset used, return true, else false.

### Q. Pair Sum Divisibility

What?

Given an array of N elements and number K, check whether array can be divided into pairs such that sum of every pair is divisible by K.

How?

- 1) For each element in array, store element%K in hashmap. If element%K is negative, store K + element%K.
- 2) Check following cases: (Loop in keyset)
  - a) If map has a key = 0, then if  $\text{map}[0] \% 2 \neq 0$ , return false.
  - b) If  $\text{map.containskey}(K - \text{val}) == \text{false}$ , return false.  
 [ val is each element in keyset, or a key of hashmap. ]
  - c) If  $K - \text{val} == \text{val}$ , check if  $\text{map}[\text{val}]$  is even, else return false.
  - d) If  $\text{map}[K - \text{ele}] \neq \text{map}[\text{ele}]$ , return false.

ex:  $N=4$ ,  $A = \{9, 7, 5, 3\}$ ,  $K=6$ .

HashMap :  $3 \rightarrow 2$ .  $\checkmark$   
 $1 \rightarrow 1$   
 $5 \rightarrow 1$ .

val : 3,  $6 - 3 = 3$ , and  $\text{map}[3] = 2$   
 which is even.  $\checkmark$

val : 1,  $6 - 1 = 5$ , and  $\text{map}[1] == \text{map}[5]$   
 so return true.

Q. Max points on a line  
what?

Given  $n$  points on a 2D plane, find maximum number of points that lie on same line.

How?

\* Slope formula :  $\frac{y_2 - y_1}{x_2 - x_1}$

- \* Create a gcd function which reduces all points as  $dy = dy/gcd$  and  $dx = dx/gcd$ .
- \* Store for this  $(dy, dx)$  pair, in a hashmap as frequency map, count of all points with this reduced  $(dy, dx)$  pair.  
i.e.:  $(dy, dx) \rightarrow$  count of points
- \* Return the max count possible for any such pair.

$N=6$

ex:  $[(1, 1), (3, 2), (5, 3), (4, 1), (2, 3), (1, 4)]$

map for  $(1, 1)$

1) $(1, 1), (3, 2)$	$dy = 1, dx = 2$	$gcd = 1 \Rightarrow (1, 2)$	$(1, 2) \rightarrow 1$
2) $(1, 1), (5, 3)$	$dy = 2, dx = 4$	$gcd = 2 \Rightarrow (1, 2)$	$(2, 1) \rightarrow 1$
3) $(1, 1), (4, 1)$	$dy = 0, dx = 3$	$gcd = 3 \Rightarrow (0, 1)$	$(1, 0) \rightarrow 1$
4) $(1, 1), (2, 3)$	$dy = 2, dx = 1$	$gcd = 1 \Rightarrow (2, 1)$	$(max here)$ $= 2 + 1$ $= 3 \text{ (I)}$
5) $(1, 1), (1, 4)$	$dy = 3, dx = 0$	$gcd = 3 \Rightarrow (1, 0)$	$(+1 \text{ because } (1, 1) \text{ is also on this dy, dn})$
6) $(3, 2), (5, 3)$	$dy = 1, dx = 2$	$gcd = 1 \Rightarrow (1, 2)$	

map for (3,2)

7)  $(3, 2), (4, 1)$

$dy = -1 \quad dx = 1 \quad gcd = -1 \Rightarrow (1, -1)$

 $(1, 2) \rightarrow 1$  $(1, -1) \rightarrow 1/3$ 

8)  $(3, 2), (2, 3)$

$dy = 1 \quad dx = -1 \quad gcd = 1 \Rightarrow (1, -1)$

9)  $(3, 2), (1, 4)$

$dy = 2 \quad dx = -2 \quad gcd = 2 \Rightarrow (1, -1) \quad \text{max} = 4.$

10)  $(5, 3), (4, 1)$

$dy = -2 \quad dx = -1 \quad gcd = -1 \Rightarrow (2, 1)$

map for (II)

 $(5, 3) \rightarrow$  $(2, 1) \rightarrow 1$ 

11)  $(5, 3), (2, 3)$

$dy = 0 \quad dx = -3 \quad gcd = -3 \Rightarrow (0, 1)$

 $(0, 1) \rightarrow 1$  $(1, -4) \rightarrow 1$ 

12)  $(5, 3), (1, 4)$

$dy = 1 \quad dx = -4 \quad gcd = 1 \Rightarrow (1, -4)$

map for (IV)

13)  $(4, 1), (2, 3)$

$dy = 2 \quad dx = -2 \quad gcd = 2 \Rightarrow (1, -1)$

 $(1, -1) \rightarrow 1/2$ 

14)  $(4, 1), (1, 4)$

$dy = 3 \quad dx = -3 \quad gcd = 3 \Rightarrow (1, -1)$

map for (IV)

15)  $(2, 3), (1, 4)$

$dy = 1 \quad dx = -1 \quad gcd = 1 \Rightarrow (1, -1)$

map for (I, III)

 $(1, -1) \rightarrow 1$  $\text{max} = 2$  (IV)

Using all maxes for each point,

(I) = 3, (II) = 4, (III) = 2, (IV) = 3, (V) = 2,

Overall max = 4

→ answer.

# We use gcd to manage quadrants.

# If class does not implement hashCode & equals,  
implement it. (Line class : int dx, int dy)Java

→ public int hashCode()

{ return this.dx + this.dy;  
}

→ public boolean equals(Object o)

{ Line l = (Line o);  
if (this.dx == l.dx && this.dy  
== l.dy)  
else return true  
return false.

## Q Morning Assembly.

what?

Given an array of  $N$  elements which contains heights of students in order which they are standing. Return minimum transfers needed to arrange the students in  $\Theta$  in order of heights.

How?

eg:  $[7 | 4 | 1 | 3 | 5 | 2 | 6]$   $N=7$

res:  $[1 | 2 | 1 | 1 | 2 | 2 | 3] \quad n$

(4-1)  
3 index

not present in  
hashmap so

(5-1)=4 present  
in hashmap

$res[5] = res[map.get(4)] + 1$

map

7 → 0      3 → 3

4 → 1      5 → 4

1 → 2      2 → 5

$max = X \neq 3$ .

$6 \rightarrow 6$

→ For each value of index check if value -1 is present in map or not. If present means this element has  $map.get(value-1)$  persons before it who are in order.

→ max count contains maximum no. of elements that are ordered.

$$\begin{aligned} \text{Minimum transfers} &= \text{array length } (N) - \text{max} \\ &= 7 - 3 = \underline{\underline{4}} \end{aligned}$$

## Q Identify Strings

What?

Given a string arrange the characters of string such that adjacent characters are distinct.

How?

- Make Frequency Map  $fmap < \text{key}, \text{value} \rangle$  where key is character of string & value is frequency of that character is value.
- If max frequency is greater than  $(\text{string.length})/2$  then string can't be arranged.

Make max heap of pair class.

Pair class  $\rightarrow$  Char c  
 $\downarrow$  int freq.

In Java.

```
public static class pair implements Comparable<pair>
```

```
int freq;
```

```
char ch;
```

```
public pair (int freq, char ch)
```

```
{ this.freq = freq;
```

```
this.ch = ch; }
```

3

```
public int compareTo (pair o)
```

{

```
if (this.freq == o.freq)
```

```
return o.ch - this.ch;
```

```
else
```

return this.freq - o.freq;

3

Priority queue will be max heap

Priority Queue < pair > pq = new Priority Queue < >  
 (Collections.reverseOrder());

In C++

struct frequency compare {

bool operator() (const pair &a, const pair &b)

{

if (a.freq == b.freq)

return b.ch < a.ch;

else

return a.freq < b.freq;

}

};

class pair {

<sup>public</sup>  
 int freq;  
 char ch;

pair () { }

pair (int freq, char ch)

{ this->freq = freq;

this->ch = ch; }

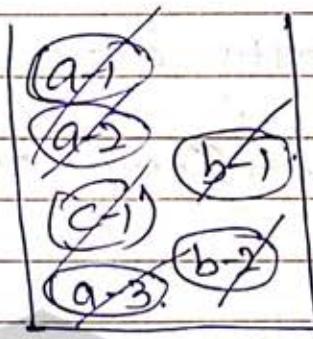
};

};

priority queue < Pair , vector < pair > , frequency map  
 > pair;

eg: aaabcb

Ans ababac



- pop 2 highest frequency.
- point chars
- push with decremented frequency.

Q Substring with similar first and last characters

what?

Count the no. of substrings of a given string which has same first and last occurrence.

How?

Make frequency map for each character

eg: a b c b d a e d b c a

a → x 2 3

count = x 2 3 5 6 8 10 12 15

b → x 2 3

18 21

c → x 2 3

when frequency

d → x 2

is 2 that is also

Ans = 21

valid substring

as per given condition

so that is why added in count.

- update frequency in map and then and in count resultant frequency

## Q Group Shifted Strings.

What?

Given an array of strings. Group all those strings together which are shifted in same way. Shifting is when we change each letter of string to its successive letters.

How?

Eg. arr = {abc, bcd, acdf, xyz, az, ba, a, z}

1) For each string calculate the gap b/w characters (adjacent) and make a string.

for eg: s = abc

$$\text{key} = *1 *2 * \leftarrow (* (b-a) * (c-a))$$

s = za

$$\text{gap} = \left( \frac{a-z}{26} \right) = (1-26) = -25$$

key = \*1

$$= -25 + 26 = 1$$

\* when gap is negative add 26.

2) If string is already present then add to the arraylist current string.

Map

\*1\*2\* → [abc, bcd, xyz]

\* → [a, z]

\*2\*2\*1\* → [acf]

\*25\* = [az, ba]

Ans

abc bcd xyz

a z

az ba

acf

All the string that belong to same shifted group will form same key string.

Q Max No. of characters between 2 same characters.

What?

Given a string find maximum no. of characters that are present b/w 2 same characters.

How?

→ Make a Hashmap to store first occurrence for each character.

At second, third... occurrences check the gap b/w first occurrence & current occurrence if greater than max update max.

eg: a b c d a b c b d a c c d f d.

	max gap	Map
at 4	$(4-0+1) = 5$	a → 0
at 5	$(5-1+1) = 5$	b → 1
at 6	$(6-2+1) = 5$	c → 2
at 7	$(7-1+1) = 7$	d → 3
at 11	$(11-1+1) = 10$	f → 13
at 14	$(14-3+1) = 12$	Ans = <u>12</u>

If no char repeats then maxlen will remain 0.

Q logger Rate limiter

What?

Given a stream of message along with timestamps each message should be printed if only if its not printed in last 10 seconds. return true if msg should be printed

How?

Make a hashmap that contains message as key. and value will be the timestamp at which it was last printed.

eg: a ✓ b ✓ c ✓ a ✗ b ✓ a ✗ c ✓ b ✓ c.  
 2 8 10 11 14 16 18 20 22  
 $11 - 10 = 1$        $14 - 8 < 10$        $20 - 8 > 10$   
 $2 > 1$  so False      point True.

Map

a → ✓ 16  
 b → ✓ 20  
 c → ✓ 22

Result → [T T T F F T F T T],

map.get(msg)

For each time stamp check (timestamp - ~~msg~~)  
 $\geq 10$  then return True and update timestamp  
 else false.

## Q) Longest Consecutive Subsequence

What?

Given an array of integers find the longest consecutive subsequence in an array.

How?

- 1) Find maximum & minimum of the array elements and Fill array elements in a set.
- 2) Make boolean array of size max+1 and contains True in all indexes.
- 3) For each num b/w min to max check if num-1 is present in HashSet or not if present then mark it false in boolean array as it can't be start of sequence.

3) For each index (num) in array (given number) check if boolean array is true. If true means this is start of subsequence. Increment length count till you find cur+1 element in set.

eg.  $\boxed{1 \ 9 \ 12 \ 11 \ 6 \ 2 \ 8 \ 3 \ 5 \ 10}$   
 0 1 2 3 4 5 6 7 8 9

boolean starts array.  $\boxed{T \ T \ F \ F \ T \ F \ T \ F \ T \ F \ T \ F \ F \ F \ F \ F \ F}$   
 0 1 2 3 4 5 6 7 8 9 10 11 12 13  
 ↓  
 2 in set

$\Rightarrow \boxed{T \ T \ F \ F \ F \ T \ F \ F \ T \ F \ F \ F \ F \ F \ F}$   
 0 1 2 3 4 5 6 7 8 9 10 11 12 13

$1 \rightarrow \text{start}[1] = \text{True} \Rightarrow \text{Count} = X \times 3 \quad \text{max} = 3$   
 (2 present (3 present)  
 inst.)

$9 \rightarrow \text{start}[9] = \text{False.} \quad \text{max} = 3$

$12 \rightarrow \text{start}[12] = \text{False.} \quad \text{max} = 3$

$8 \rightarrow \text{start}[8] = \text{True} \Rightarrow \text{Count} = X \times 3 \times 4 \times 5 \quad \text{max} = 5$

$5 \rightarrow \text{start}[5] = \text{True} \quad \text{Count} = 2 \quad \text{max} = 5$

ans = 5

\* we are finding the starts of subsequences and checking the length from that.

## Heap Based Questions

Q. IPO

what?

Help pepcoding design the best way to maximize its total capital after finishing at most  $k$  distinct projects. Each project's price as profit and capital needed to start it is given. Initial Capital is  $W$ . Return maximised final capital.

How?

1. Sort on the basis of Capital in ascending order.
2. Sort on the basis of Profit in descending order if capital is same (used in priority queue).

ex: 

	a	b	c	d	e	f	
C:	80	150	90	200	250	350	, $W = 100$
P:	40	60	50	70	50	300	, $k = 4$ .

After sorting: a b c d e f  
 C: 80 90 150 200 250 350  
 P: 40 50 60 70 50 300.

3. Now priority queue uses capital to perform profit comparison, initially priority queue has all elements.
4. Now one by one elements are added in priority queue after checking if its capital  $\leq$  the capital we currently have.

$k=1$  I)  $a \frac{6}{}$   
 $W=100$   $\frac{80 \quad 90}{40 \quad 50}$   
 added in 1st pass.

profit in this  $\Rightarrow c$  gives more capital so  
 c is removed & its profit  
 capital is added in answer.

II)  $W = 150 \quad (100 + 50)$

$k=2$

a	b
50	150
40	60

$\Rightarrow b$  gives more profit.

III)  $W = 210 \quad (150 + 60)$

$k=3$

a	d
80	200
40	70

$\Rightarrow d$  gives more profit.

IV)  $W = 280 \quad (210 + 70)$

$k=4$

a	e
80	250
40	50

$\Rightarrow e$  gives more profit.

$\therefore$  Total profit earned = 330 using by doing c, b, d, e projects.

Implementing comparators in C++.

class Project{

public:

int c;

int p;

Project ( int p, int c ) {

this  $\rightarrow$  p = p;

this  $\rightarrow$  c = c;

}

Project () {

// default constructor is needed

}

// to create objects

};

bool CapitalComparator( Project &a, Project &b ) {

return a.c < b.c;

};

PEPCODING - PURSUIT OF EXCELLENCE AND PEACE

```

struct ProfitComparator {
    bool operator () (const Project &a, const Project &b)
    {
        return a.p < b.p;
    }
};

```

In main, create vector<Project> v and after inserting all Project pairs in it, sort it as:

- sort (v.begin(), v.end(), CapitalComparator);

Now create a priority queue as:

- priority - queue<Project, vector < Project >, ProfitComparator > pq;

### Implementing comparators in Java

```

public static class Project {
    int p;
    int c;
    Project (int p, int c)
    {
        this.p = p;
        this.c = c;
    }
}

```

```

public static class profitComparator implements
    Comparator<Project> {
    public int compare (Project p1, Project p2) {
        return p1.p - p2.p;
    }
}

```

```

public static class capitalComparator implements
    Comparator<Project> {
    public int compare (Project p1, Project p2)

```

{ return pl.c - p2.c }  
 ?  
 ?

In main, sort array in which you insert all elements as Project pairs as:

→ Arrays. sort (projects, new capacityComparator());  
 ↓  
 projects is the array.

After this, create priority queue as:

→ Priority Queue <Project> pq = new Priority Queue  
 <>(Collections. reverseOrder (new profitComparator()));

Q. Print k smallest elements in an array in given order.

What?

Print first k smallest elements in an array in the order in which they appear in the array.

How? ex: {4, 2, 7, 11, 9, 15, 7, 6, 3, 8, 20, 19}, k=4.

→ Create a priority queue of size k of max type.

I)	<table border="1"><tr><td>4</td><td>2</td><td>7</td><td>11</td><td>11</td></tr></table>	4	2	7	11	11	4, 2, 7, 11
4	2	7	11	11			
II)	<table border="1"><tr><td>4</td><td>2</td><td>7</td><td>9</td><td>9</td></tr></table>	4	2	7	9	9	9
4	2	7	9	9			
III)	<table border="1"><tr><td>4</td><td>2</td><td>7</td><td>9</td><td>9</td></tr></table>	4	2	7	9	9	15
4	2	7	9	9			
IV)	<table border="1"><tr><td>4</td><td>2</td><td>7</td><td>7</td><td>7</td></tr></table>	4	2	7	7	7	7
4	2	7	7	7			
V)	<table border="1"><tr><td>4</td><td>2</td><td>7</td><td>6</td><td>7</td></tr></table>	4	2	7	6	7	6
4	2	7	6	7			
VI)	<table border="1"><tr><td>4</td><td>2</td><td>6</td><td>3</td><td>6</td></tr></table>	4	2	6	3	6	3
4	2	6	3	6			
VII)	<table border="1"><tr><td>4</td><td>2</td><td>6</td><td>3</td><td>6</td></tr></table>	4	2	6	3	6	8, 20, 19
4	2	6	3	6			

Now, for printing, if array element < peek, print, and if it is == peek, remove (so that peek can change).  
 → 4, 2, 6, 3 is ans.

Q. Print kth smallest fraction.  
what?

You will be given an array of prime numbers and a number k, find kth smallest fraction  $n/d$  such that both belong to array of prime numbers.

How 2 ex: [ 3 5 7 2 11 ]

All cases:  $\frac{2}{11} \quad \frac{3}{11} \quad \frac{5}{11} \quad \frac{7}{11}$

$\frac{2}{7} \quad \frac{3}{7} \quad \frac{5}{7}$

$\frac{2}{5} \quad \frac{3}{5}$

$\frac{2}{3}$

If the array is not sorted, sort the array to only make pairs in the way shown above -

→ Now simply apply a comparator which sorts fraction pairs : -

C++

```
→ bool FractionComparator (fpair &a, fpair &b) {
    return (a.n*(1.0)/a.d) < (b.n*(1.0)/b.d);
}
```

\* Now return k<sup>th</sup> smallest fpair with numerator = n & denominator = d .

Java Implement compareTo in class.

```
→ public int compareTo (fpair o) {
    double val1 = this.n*(1.0)/this.d;
    double val2 = this.o. n*(1.0)/o.d;
```

```

if (val1 < val2) {
    return -1;
} else if (val1 == val2) {
    return 0;
} else {
    return 1;
}

```

### Q. Top K Frequent Elements

What?

Given a non-empty array of integers, find top K most frequent elements.

How? ex: N = 6

{1, 1, 1, 2, 2, 3}  
k = 2

1. Make a frequency map.
2. Insert frequency map pairs in priority queue sorted on the basis of frequency. (Pair with frequency greater are on top).

PQ /      (1-3)      (2-2)      (3-1)  
Vector

3. Now pop and print values of first k pops.  
Ans: [1, 2]

# Similar question: Top K Frequent Words.  
[numbers replaced by words; rest everything is exactly same.]

Q. Height of heap  
what?

You will be given a binary heap of size  $N$ ,  
return height of heap.

How?

# Use log function to calculate height of heap.

↓  
log floor method used.

for 8 : ans = 3 ( $2^3 = 8$ ) .

// Code

```
int solve(int N)
{
    int x = 0;
    while ((1 << x) <= N) {
        x++;
    }
    return x - 1;
}
```

Returns Floor as for all 4, 5, 6, 7 it should give  
floor height as 2; but for other levels..

Q. kth largest element in a stream.

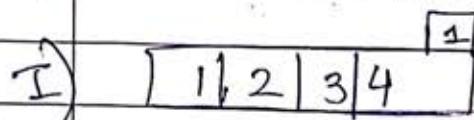
what?

You are given input stream of  $m$  integers, find  
kth largest element for each element in system  
stream.

Now? ex: {1, 2, 3, 4, 5, 6},  $k = 4$ .

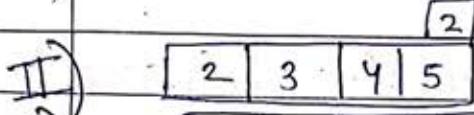
Simply create a ~~min~~ priority queue of  $k$  size,  
if ~~elements~~ pt. you have not inserted  $k$  elements,  
answer is  $-1$ , else, return the peek if  $k$ th

is being inserted, else, if element peek, remove  
~~else~~ peek element (pop), add current element  
 and now return peek.

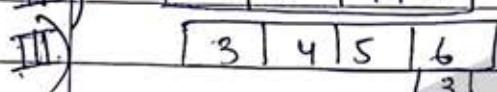


for 1, 2, 3 → ans = -1, -1, -1

for 4 → ans = 1



for 5 → ans = 2



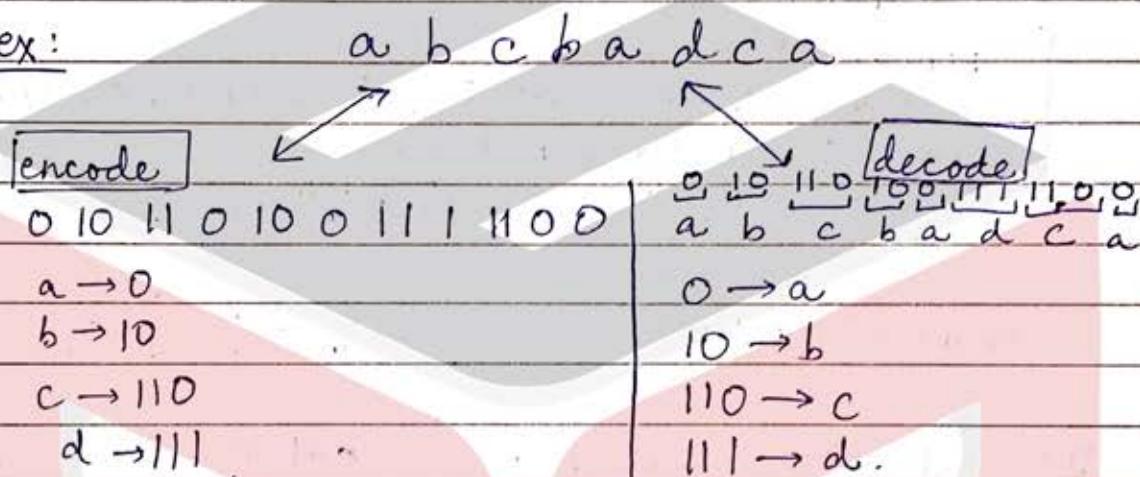
for 6 → ans = 3.

ans ⇒ [-1, -1, -1, 1, 2, 3]

## Huffman Encoding and Decoding

- a) You will be given a string, and you will have to decode it into binary : decoding.
- b) You will be given a binary string, and you will have to encode it into a character string : encoding.

ex:



Design goals to be implemented:

- 1) Each character has different encoding (no prefixes common).
- 2) Character which occurs more should have less number assigned to it, i.e.; it should be given a shorter binary code.

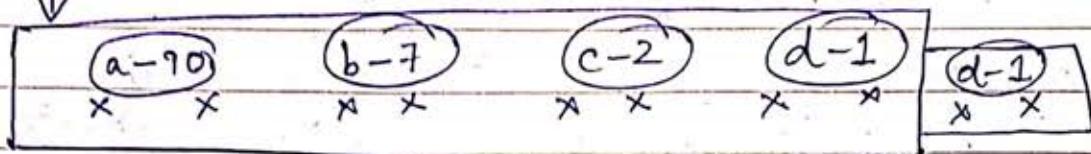
ex: a → 90, b → 7, c → 2, d → 1

Algorithm

- 1) Feeder: A feeder function tells you frequency of each character; Parses string to frequency map. 
$$\begin{bmatrix} a \rightarrow 90 & c \rightarrow 2 \\ b \rightarrow 7 & d \rightarrow 1 \end{bmatrix}$$

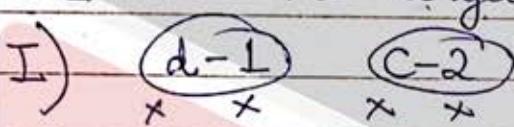
2) Frequency Map  $\rightarrow$  Priority Queue of Nodes where each Node has value as char-freq, and left and right as NULL initially.

$\downarrow PQ$

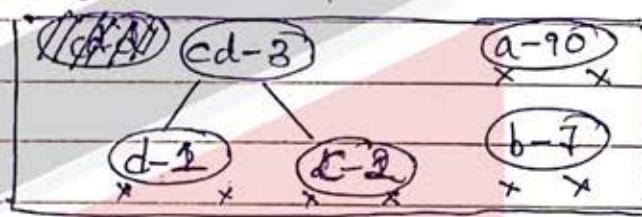


3) while (pq.size() > 1),

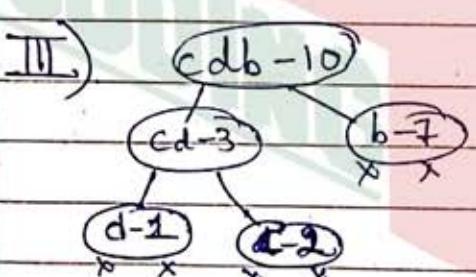
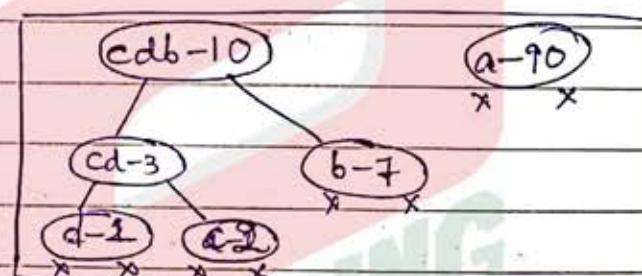
[remove 2 nodes.  
merge into 1, root is both data + both values.  
add the merged node back.]



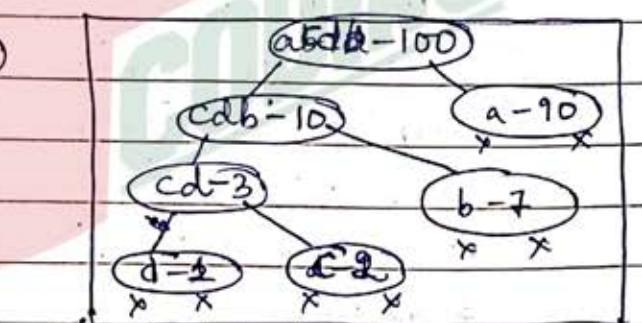
data = cd, val = 3



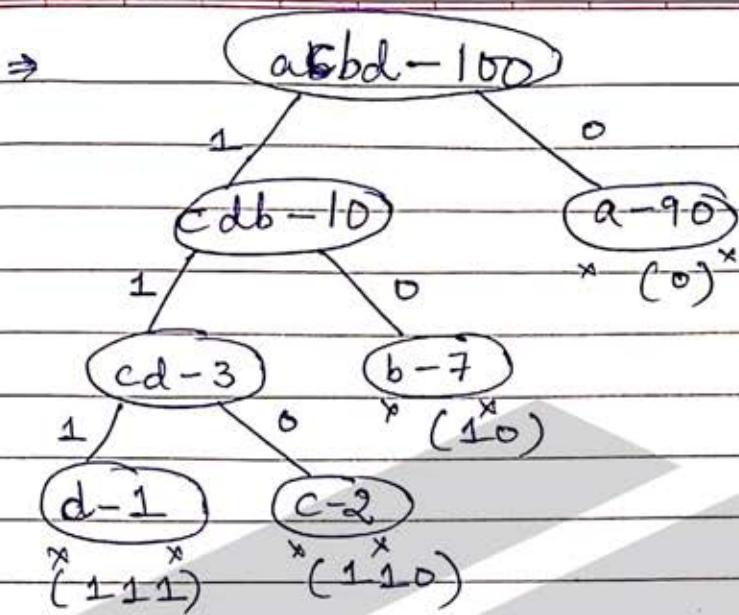
data = cdb, val = 10



data = abdd, val = 100



Now use this tree to create codes for each character at leaf node. Since left has all smaller, left calls add 1 in string, right calls add 0 in string.



∴ mapping done as:

$a = 0$   
 $b = 10$   
 $c = 110$   
 $d = 111$

→ The mappings as we see meet all design goals, because since we use min priority queue, the characters used least have more depth and longer codes.

Also since mapping is done at leaf nodes, which contain only characters, we have no common prefixes.

## BIT Masking Based Questions

Q Find the Difference

What?

Given 2 strings find all the characters which is different from both strings

How?

XOR all the characters of string 1 and string 2  
last remaining character is different character

eg:  $s_1 = a \ b \ c \ d \ e$   
 $s_2 = e \ c \ b \ d \ a \ g.$

$$\begin{aligned} \text{XOR} &= a \wedge b \wedge c \wedge d \wedge e \wedge e \wedge c \wedge b \wedge d \wedge a \wedge g \\ &= (a \wedge a) \wedge (b \wedge b) \wedge (c \wedge c) \wedge (d \wedge d) \wedge (e \wedge e) \\ &= 0 \wedge g \\ &= g. \end{aligned}$$

$a \wedge a = 0$
$a \wedge 1 = a$ (Complement)
$a \wedge 0 = a$

Q Find the Odd occurrence

What?

Given an integer array find the element with odd occurrence. (only element like this)

How?

XOR all the elements of array last remaining is the result.

## Q Uncommon Characters

What?

find and print uncommon characters of two given strings. Here uncommon character means that either the character is present in one string or it is present in other but not in both. (string has only lowercase chars).

How?

- For each string make a bitmask (for 26 alphabets)  
If the string contains that alphabet set  
the bit otherwise it is set to 0.

eg: s1: a b c d d f g p r  
s2: g h i a n d f p.

mask 100000000011000000001101111

`mask: 0000000001010000 0111101001`

xor 0000000010010000110000110  
v n in cb

An bching

## why

If both have characters then  $\text{XOR}(1 \wedge 1)$  is 0  
result won't contain that character :

- \* Set Bit  $X = X | (1 \ll K)$  (To set  $K^{th}$  bit)
- \* Unset Bit  $X = X \& (\sim (1 \ll K))$
- \* Toggle Bit  $X = X \wedge (1 \ll K)$

## Q Is-Sudoku Valid

what?

Given a  $9 \times 9$  board of Sudoku. Validate the filled cells. Each row <sup>each</sup> must contain digits 1 to 9 without repetition & each of the 9  $3 \times 3$  subboxes must contain the digits 1-9 without repeat.

how?

	c <sub>0</sub>	c <sub>1</sub>	c <sub>2</sub>	c <sub>3</sub>	c <sub>4</sub>	c <sub>5</sub>	c <sub>6</sub>	c <sub>7</sub>	c <sub>8</sub>
r <sub>0</sub>	5	3	.	.	7	.	.	0	0
r <sub>1</sub>	6	0	0	2	9	5	0	9	0
r <sub>2</sub>	0	9	8	0	0	0	0	6	0
r <sub>3</sub>	8				6				3
r <sub>4</sub>	4			8		3			1
r <sub>5</sub>	7			2					6
r <sub>6</sub>		6				2	8	0	
r <sub>7</sub>				4	1	9	.		5
r <sub>8</sub>				8			7	9	

We will use bitmask do store which no.(digit) is used in this row/column/box.

integer row[9] → contains mask for each row  
 integer col[9] → contains mask for each column  
 integer sn[3][3] → contains mask for each box

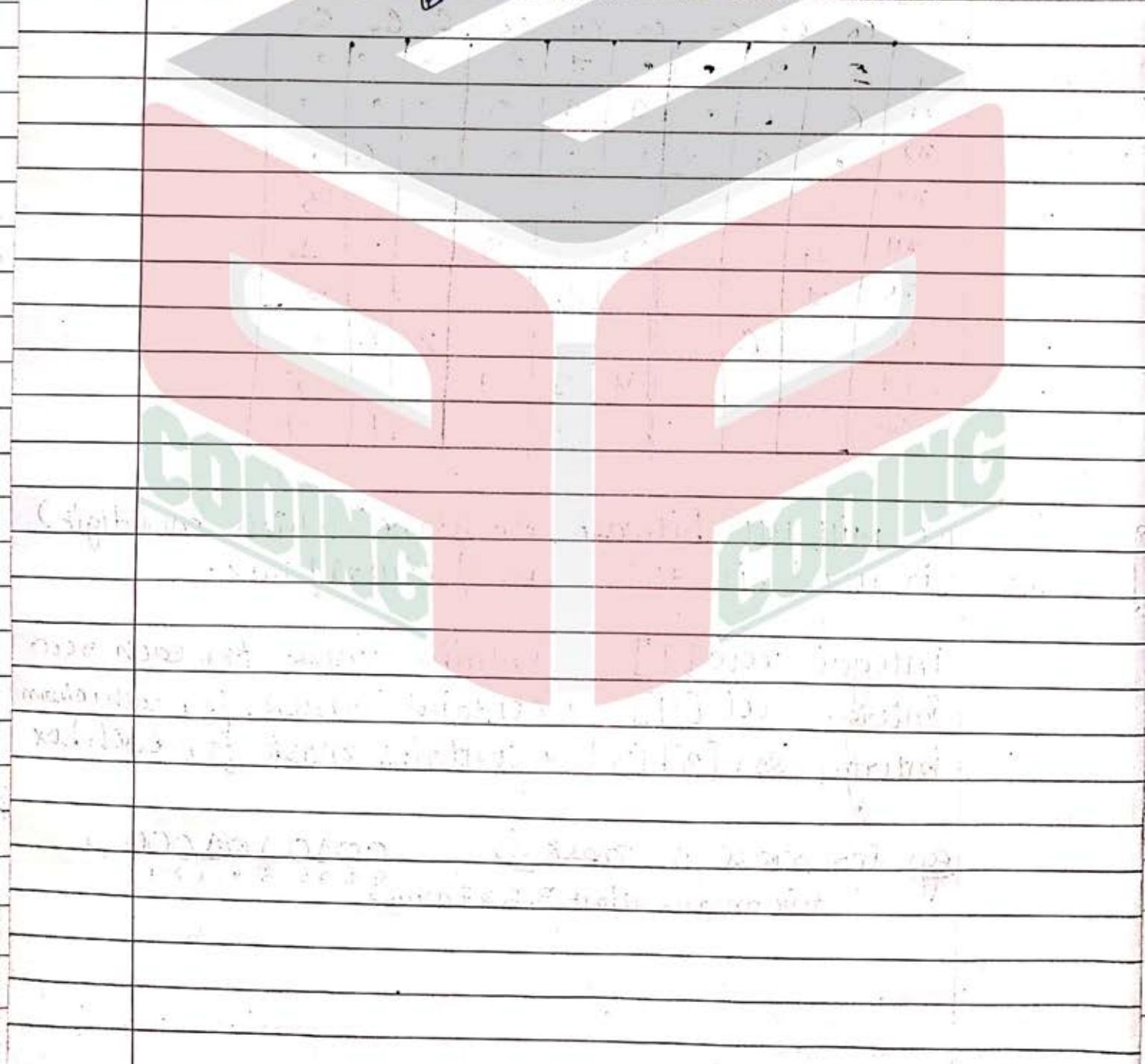
eg: For row 1 mark is ~~0010101000~~  
 this means digit 3, 5, 8, 7 are used

For every box 3 conditions are checked following  
 $i, j$  are row & col no.

- $\text{row}[i] \& (1 << (\text{digit})) \neq 0$  (already present)
- $\text{col}[j] \& (1 << (\text{digit})) \neq 0$  (so false)
- $\text{sum}[i/3][j/3] \& (3 << \text{digit}) \neq 0$  (already present in box)

If all checks fail ie box is valid.

Set bit true ~~in~~ in all three masks.



### A. Meeting Rooms What?

You are given start time and end time of various meetings where  $s_i < e_i$  for each one. Return minimum number of rooms required to hold these meetings.

How?

Approach 1: left-right.

Sort the two ~~sort~~ start times and end times array. Keep left on start[0] and right on end[0]. While comparing, if  $\text{start}[0] \leq \text{end}[i]$ , count++, else count--(if  $\text{end}[i] < \text{start}[i]$ ). Answer is maximum <sup>count</sup> at any particular i.

ex:  $\rightarrow \downarrow \downarrow$   
 starts: 2, 3, 4, 5, 8, 10, 12, 13, 14, 15, 16, 17, 18  
 ends: 6, 6, 7, 7, 9, 11, 13, 14, 15, 18, 19, 20, 20, 20

count = X X ~~X~~ 4 ~~X~~ X X X X ~~X~~ X X ~~X~~ X ~~X~~   
~~X~~ 4 (5) 4 3 ~~X~~ X 0

→ maximum count at any moment was 5.

# If you first decrement in case of ==, ans is 4

Approach 2: Create an array of size = max value of all starts and ends array.

Put +1 at all indices of starts[i] in array, and -1 and all ends[i]. Answer is maximum prefix sum at any point.

		1	1	1	1	-2	-1	1	-1	1	-1	1	10	10	10	1	1	10	-1
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
psum:	1	2	3	(4)	2	1	2	1	2	1	2	2	2	3	4	4	-3	20	0

ans