

Agenda

→ Java / Python / C / C++

STL

Macros

Design our CP template

OOPS (Classes)

Structs (C++)

→ DS [C++]

→ Pointer

→ It is much faster than java / python / C# etc

→ C++ has a rich STL library
Standard
Template
Library }

→ The CP editorials , mostly are in C++

→ You can write very short code when compared with Java

\Rightarrow string \rightarrow $\{ \}$
 \searrow
curly braces

min operation to make it
balanced.

all sequences are of even length

$\} \{ \rightarrow \underline{\underline{2}}$
 $\{ \}$
 $\{ \}$ $\rightarrow \underline{\underline{2}}$

$\{ \{ \{ \{$
 $| \{ \{ \} \}$ $\rightarrow \underline{\underline{2}}$

$\{ \{ \{ \{ \{ \{$ \rightarrow balanced
 $\{ \{ \{ \{ \{ \}$ $\rightarrow 2 + 2$
 $\{ \{ \{ \} \}$ $\rightarrow \underline{\underline{4}}$
 $\{ \{ \{ \} \}$

$\overbrace{\{ \{ \{ \{ \}$
 $\{ \{ \{ \}$

$\rightarrow \underline{\underline{2}}$

$\{ \{ \{ \{ \}$
 $\{ \{ \{ \}$

$\rightarrow \underline{\underline{\{ \{ \{ \{ \}$

Calculate the count of unbalanced opening & closing braces.

total length even \rightarrow

$\overbrace{\{ \{ \{ \{ \}$	$\overbrace{\{ \{ \{ \{ \}$
$\underbrace{\{ \{ \{ \}$	$\underbrace{\{ \{ \}$
2 +	

we can remove the balanced part whenever
for an opening brace we get an immediate
closing brace.

(Stack)

\swarrow \swarrow
~~3~~ ~~3~~ ~~3~~ { { { } } }
 \nearrow \nearrow \nearrow

#include <stack>

closing = ~~0~~ ~~1~~ ~~2~~ 2

3
3
3

$$\begin{aligned}
 \text{Ans} &= \\
 &2 + 1 + 1 \\
 &\rightarrow \underline{\underline{4}}
 \end{aligned}$$

Q₂ Given a stream of lowercase alphabets
 for each new alphabet check who is the first
 non repeating character ??

↪ a b b c a
 ↓ ↓ ↓ ↓ ↓
 a a a a c ...

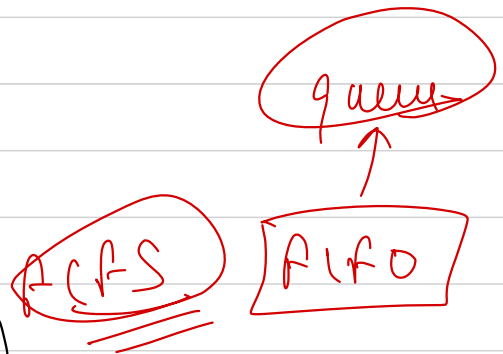
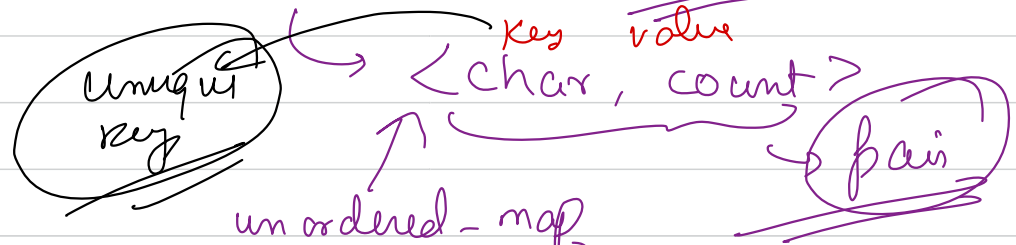
↗ character →
 n → $n \leq 10^6$

Test Case 2

↪ g → g
 g → -
 h → h
 m → h

Q ↳ When does a char loses the chance of being non-repeating??

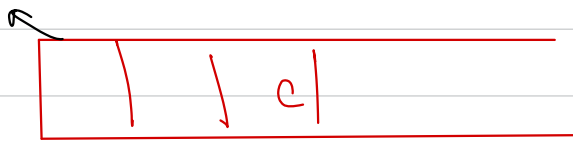
ans \rightarrow The first time it repeats \rightarrow (1)



Dict / HashMap
insertion - $O(1)$
deletion - $O(1)$
search - $O(1)$

Q If we have multiple char with freq 1, who is the first non-repeating??
The one that came first

#include <queue> queue → 26



Dict

unique

$O(1)$ insert
 $O(1)$ delete
 $O(1)$ search

$O(n)$

Key	value
char	count
'a'	2
'b'	2
'c'	1

q →
 'a', 'b', 'c', 'b', 'a'
 ↓ ↓ ↓ ↓ ↓
 a a a a c

26

#include <unordered_map>

SC → $O(1)$

TC → $O(1)$

per query

Hardy

Q-3 fence

$$4 + 4 - 1 + 3 + 4 - 2$$

2

	0,1				
	1,1				
			P		
			P	P	P
	P				

x, y

ump

$\langle \text{int}, \langle \text{int}, \text{int} \rangle \rangle$

$\langle x, \langle y, 1 \rangle \rangle$

$(a, b) \rightarrow$ do we have a plant or not?

matrix \rightarrow $O(n \times m)$

ump

$\langle \text{int}, \text{ump} \langle \text{int}, \text{int} \rangle \rangle$

↑ ↑

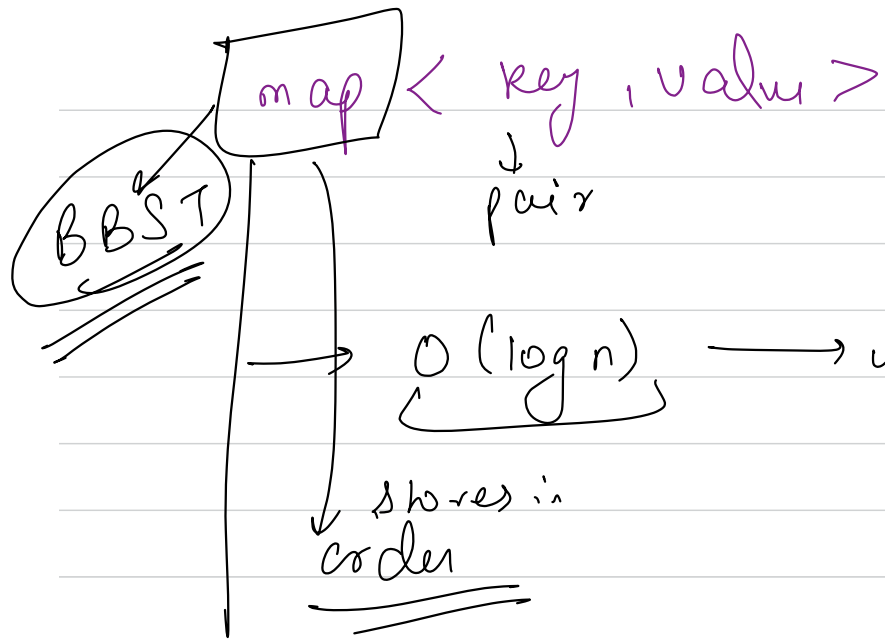
x ~~y~~

1

Q⁷⁷ Can I initialise an unordered map as

$\text{unordered_map} \langle \text{pair} \langle \text{int}, \text{int} \rangle, \text{int} \rangle$

1
randomly



map < int, int > m;

m[1] = 2

manhattan

Q2

$dist(p_1, p_2)$

$$\rightarrow |x_1 - x_2| + |y_1 - y_2|$$

$$\rightarrow \underline{\max(x_1 - x_2, x_2 - x_1)} + \underline{\max(y_1 - y_2, y_2 - y_1)}$$

$$p_1 \rightarrow x_1, y_1$$

$$\rightarrow \max((x_1 + y_1) - (x_2 + y_2), (x_1 - y_1) - (x_2 - y_2),$$

$$p_2 \rightarrow x_2, y_2$$

$$(-x_1 + y_1) - (-x_2 + y_2), (-x_1 - y_1) - (-x_2 - y_2))$$

$$f_1(p_1) = x_1 + y_1$$

$$f_1(p_2) = x_2 + y_2$$

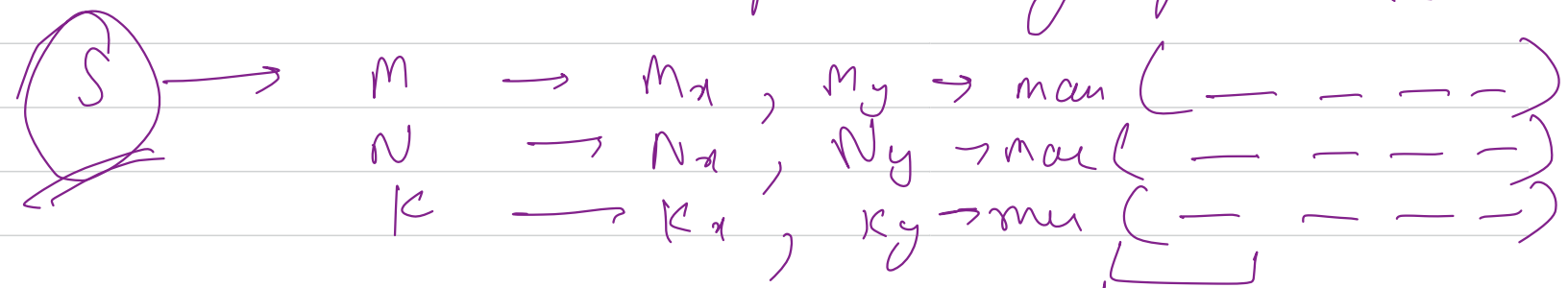
$$f_2(p_1) = x_1 - y_1$$

$$f_3(p_1) = -x_1 + y_1$$

$$f_4(p_1) = -x_1 - y_1$$

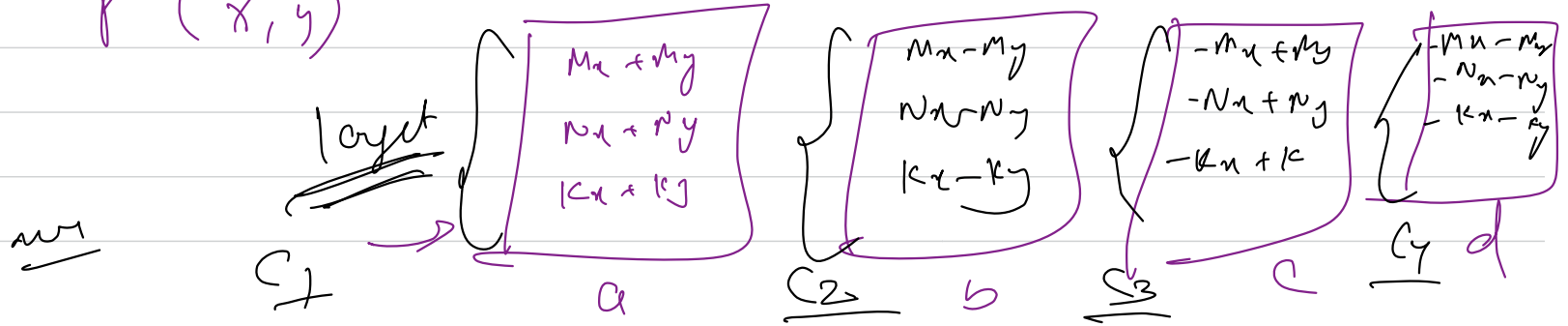
→
→
→

we will be given a point P , & we want to calc
 max manhattan dist from any point in S .



over $\boxed{\quad}, \boxed{\quad}, \boxed{\quad}, \boxed{\quad}$

$f(x, y)$



Priority Queue

→ ? → $O(\log n)$

+ → $O(\log n)$

- { ?

visited array

<https://github.com/singhsanket143/CppCompetitiveRepository/blob/master/UnacademyFreeLiveClass/SeinField.cpp>

<https://github.com/singhsanket143/CppCompetitiveRepository/blob/master/UnacademyFreeLiveClass/firstnonrepeatingcharacter.cpp>

<https://github.com/singhsanket143/CppCompetitiveRepository/blob/master/UnacademyFreeLiveClass/fence.cpp>
