

String Algorithms - I





How Strings are created in Java?

```
// Most common, short way
String str1 = "Hello World";

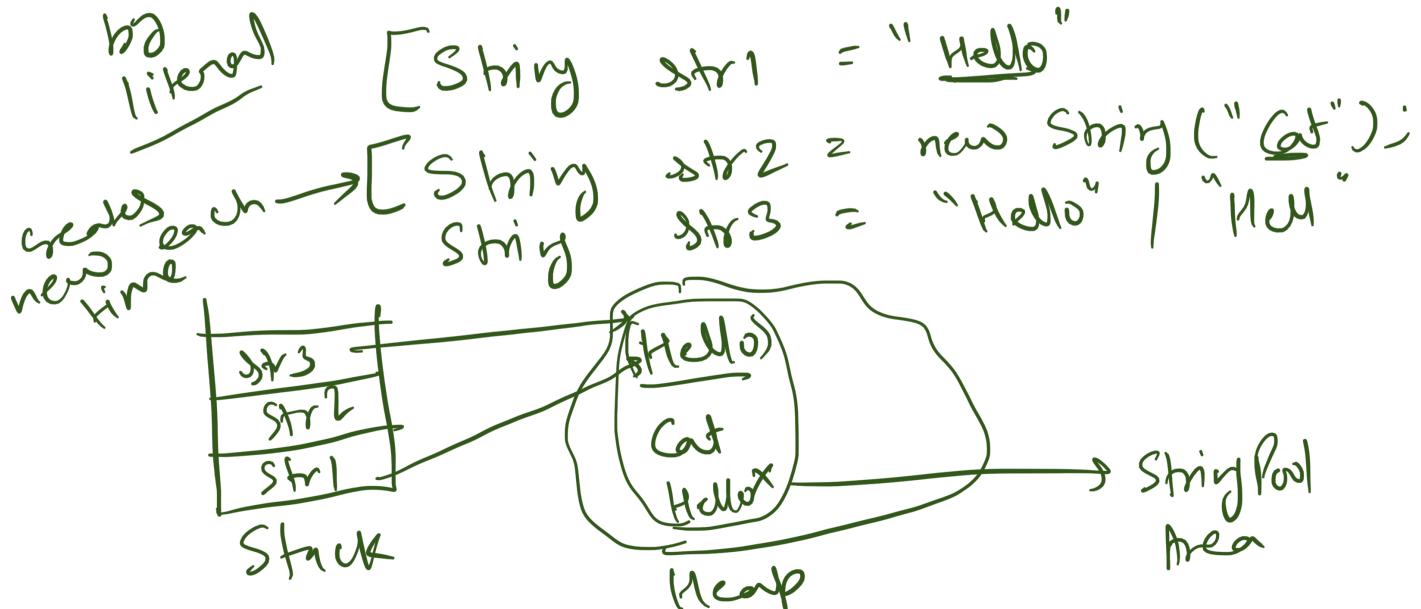
// Using the `new` keyword and passing text to the constructor
String str2 = new String("Hello World");

// Initializing an array of characters and assigning them to a String
char[] charArray = {'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd'};
String str3 = new String(charArray);
```

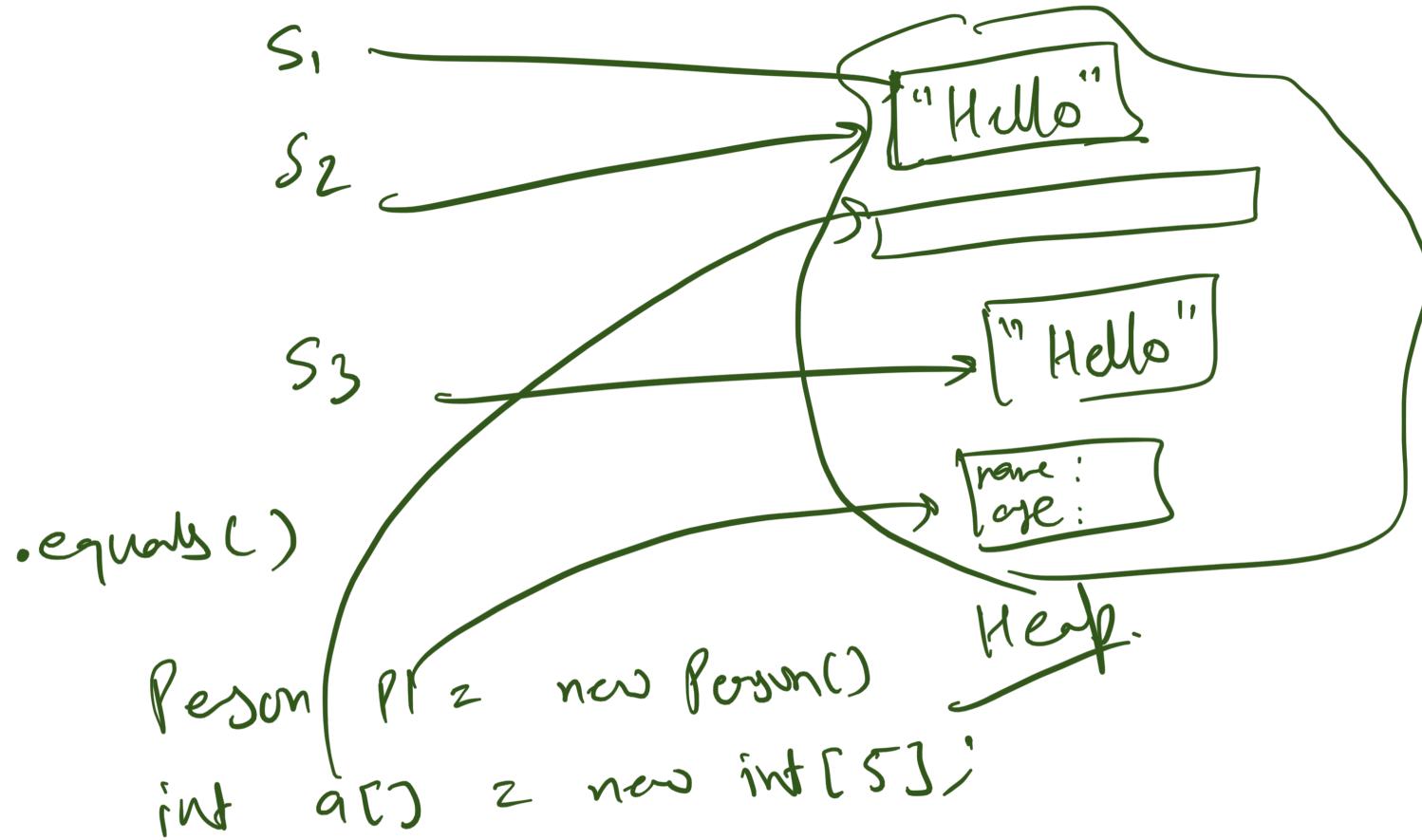
Strings are Immutable in Java

Optimisation

How Strings are stored in Java?



=
new



String Methods

indexOf()

contains()

substring()

✓ trim()

✓ isEmpty()

"Hello"

True / false.

"Robin hood"

0 4 S

indexOf('o') → 4

"ell"

→

s1.substring(0, 5) → Robin

↑
exclusive

And so on...



More can be found here: <https://www.programiz.com/java-programming/library/string>

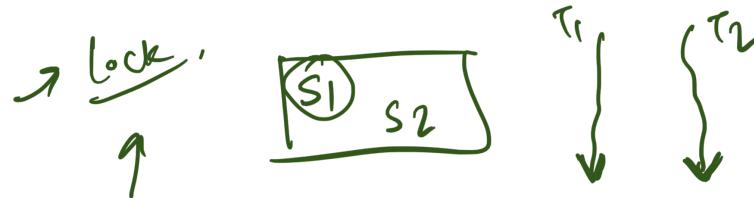
StringBuilder & StringBuffer

Not T.S

Thread Safe

Methods:

- append() ✓
- insert() ✓
- replace() ✓
- delete() ✓
- reverse() ✓



	String	StringBuffer	StringBuilder
Mutable	No	Yes	Yes ↑
Thread-Safe	Yes ✓	Yes ✓	No ═
Time Efficient	No	No ✓	Yes ↑
Memory Efficient	No	Yes	Yes ↑

Check if Two Strings are Anagram of each other.

Anagram : Two string with same types of character, repeating same number of times /
Permutations of a string are anagram.

"hello" , "olhed"

True.

"abc"
↳

bac , cb a
bc a -

"cod", "Dog" → Yes
"Cat", "Act" → Yes
"ball", "bat" → No

String s1 = "hello"

String s2 = "olhel" →

Space → char[] c1 = s1.toCharArray()

Time → Arrays.sort(c1);

n → $O(n \log n)$

toCharArray()

['h', 'e', 'l', 'l', 'o']

↓ Sort → $n \log n$

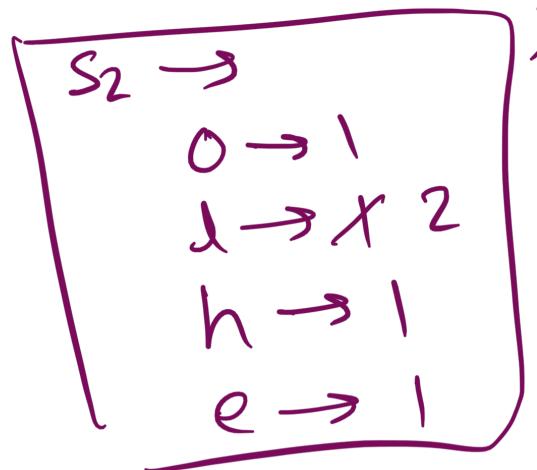
→ ['e', 'h', 'l', 'l', 'o']

→

HashMap

$s_1 \rightarrow h \rightarrow x^0$
 $e \rightarrow x^0$
 $l \rightarrow y z x^0$
 $o \rightarrow x^0$

map.



"123"
"321"

$s_1 \rightarrow "h\$@12bb\$"$
 $s_2 \rightarrow "1@2bh\$b\$"$

Space

Time $\rightarrow \underline{O(N)}$

ASCII.

256 int

$a[] \rightarrow$

0	2	0	0	0	10	10	10	2	10	10
a	b	c	d	e	h	i	2	\$	@	

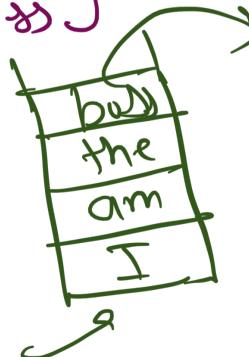
Program

Reverse the words in a given String

"I am the boss" → [I
am
the
boss]

"boss the am I"

- ① `split(" ")` → Space.
- ② Stack. uses Space.
- ③ without any extra space.



$S_1 \rightarrow$ "cat" \rightarrow "cat"
"the cat" \rightarrow "cat the"

① Reverse every word

"I ma eht ssob"

"I am the boss"

② Reverse the whole string

"boss the am I"

Practice Problems

1. Check if a String is a subsequence of another string.
 - a. abbc is subsequence of paabcbbcd
 - b. abc is not subsequence of baacd
2. Find the Leftmost character that repeats in a String.
 - a. p is the leftmost character that repeats in applejuice
3. Find the length of the last word in a given String
 - a. The length of the last word in “I am the boss” is 4 (boss)