(9) → Implement strStr() – <indexOf>

① → Question –

Return the first occurrence of needle in haystack, or -1 if needle is not part of haystack

→ needle is an empty string? → return 0

② → Examples –

haystack = "hello" , needle = "ll"

→ 2

haystack = "aaaaa", needle = "bba"

→ -1

haystack = "" , needle = ""

→ 0

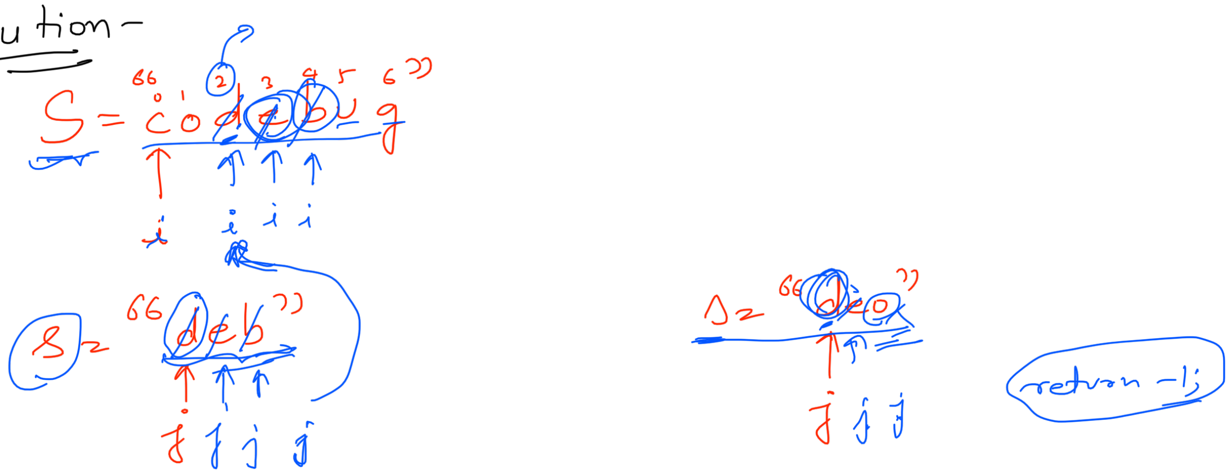
③ → Intuition –

S = "codebugisawesome"

s = "bug"

Run through entire s to see if s is there in S if first character matches...

④ → Solution –



One optimisation-

S = "$\overset{0}{c}\overset{1}{o}\overset{2}{D}\overset{3}{e}\overset{4}{B}\overset{5}{u}\overset{6}{g}$"

$\uparrow$ i

cI = 2

s = "debug"

$\uparrow$ j

len = 5

5

$2+5-1 \Rightarrow 6$

sI = 2 lI = 6

[2, 3, 4, 5, 6] ✓

cI = 3

$3+5-1 \Rightarrow 7$ (not there in S) we break now ... Done

$\underbrace{idx + len(s) - 1}_{} > \underbrace{len(S) - 1}_{\text{last index of S}}$

$idx + len(s) > len(S)$

break

5 → Time/Space Complexity-

$T = O(len(S) \cdot len(s))$

$S = O(1)$ ✓

H = "abcdefghi"

N = "abcd"

H.N

6 → Additional Info-

KMP - Will be discussed in a medium/hard question

7 → Code walk through-

```
int strStr(String S, String s) {
        // Condition
        if (s.length() == 0) {
                return 0;
        }
        //when needle is bigger than haystack
        if (s.length() > S.length()) {
                return -1;
        }
int i=0;
while( i< S.length()) {
    if(((i)+(s.length())-1)) >
        (S.length()-1)) {

        break;
    }

    int j=0;
    while(j< s.length()) {
        if(s.charAt(j) != S.charAt(i+j)){
            break;
        }
        j+=1;
    }
    if( j== s.length()) {
        return i;
    }
    i+=1;
}
return -1;
}
```

a~~pple~~ ②

~~ple~~ = 3

j=0 j=~~2~~3