1.) String has all unique characters

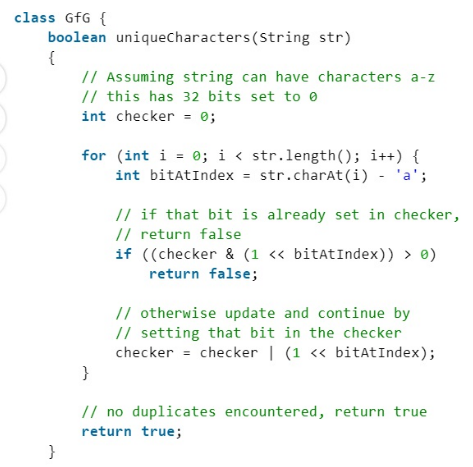
→ You can not use extra space or additional data structures.

Smdasp → false

$n + n-1 + n-2 + \dots = O(n^2)$ space → $O(1)$

matching & comparing is brute force.

→ We can try it with Hashset



```
class GfG {
    boolean uniqueCharacters(String str)
    {
        // Assuming string can have characters a-z
        // this has 32 bits set to 0
        int checker = 0;

        for (int i = 0; i < str.length(); i++) {
            int bitAtIndex = str.charAt(i) - 'a';

            // if that bit is already set in checker,
            // return false
            if ((checker & (1 << bitAtIndex)) > 0)
                return false;

            // otherwise update and continue by
            // setting that bit in the checker
            checker = checker | (1 << bitAtIndex);
        }

        // no duplicates encountered, return true
        return true;
    }
```

String = dadt

bitindex.

3 → (0 & 1000) false

⇒ checker = 0 | 1000

1000

⇒ 0 → (1000 & 1) > 0

0

checker = 1000 | 1

1001 & 1000 = 1001

> 0 false