

Resursion

what about the stack frames

```
/**  
 * Eliminates duplicate chars from a String  
 * E.g.: deduplicate("abcbaabc") is "abc"  
 */
```

```
fun deduplicate(s: String): String = run {  
    fun dedup(s: String, acc: String): String = when {  
        s.isEmpty() -> acc  
        s.first() in acc -> dedup(s.drop(1), acc)  
        else -> dedup(s.drop(1), acc + s.first())  
    }  
    dedup(s, "")  
}
```



















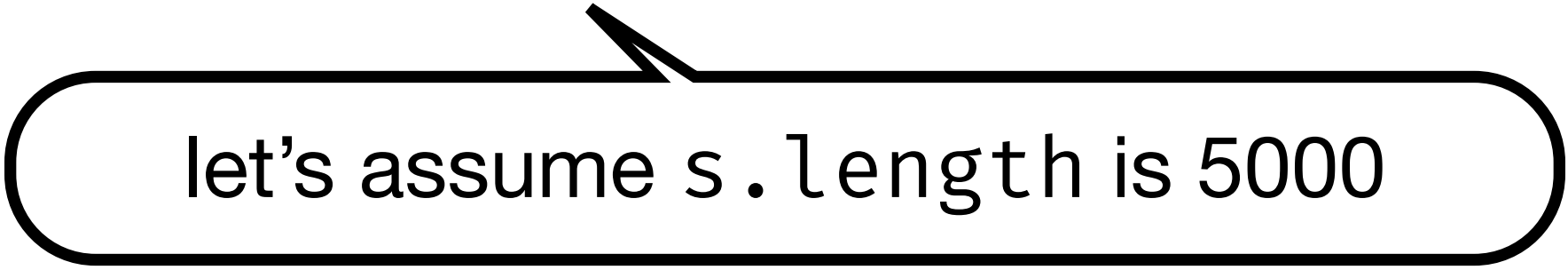












let's assume $s.length$ is 5000

Exception in thread "main"
java.lang.StackOverflowError

deduplicated(

Recursion

watch out the stack frames

```
/**
 * Eliminates duplicate chars from a String
 * E.g.: deduplicate("abcbaabc") is "abc"
 */
fun deduplicate(s: String): String = run {
    fun dedup(s: String, acc: String): String = when {
        s.isEmpty() -> acc
        s.first() in acc -> dedup(s.drop(1), acc)
        else -> dedup(s.drop(1), acc + s.first())
    }
    dedup(s, "")
}
```

deduplicate(s)

*Exception in thread "main"
java.lang.StackOverflowError*

let's assume s.length is 5000

Recursion

tail recursion

```
/**
 * Eliminates duplicate chars from a String
 * E.g.: deduplicate("abcbaabc") is "abc"
 */
fun deduplicate(s: String): String = run {
    tailrec fun dedup(s: String, acc: String): String = when {
        s.isEmpty() -> acc
        s.first() in acc -> dedup(s.drop(1), acc)
        else -> dedup(s.drop(1), acc + s.first())
    }
    dedup(s, "")
}

deduplicate(s)
```