|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | l | o | n | g | e | s | t |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| s | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| t | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| o | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2 |
| n | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 |
| e | 0 | 0 | 1 | 2 | 2 | 3 | 3 | 3 |

O/P - o n e

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | l | o | n | g | e | s | t |
|  | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | |
| s | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | s | | |  | | --- | | s | |
| t | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | "" | | |  | | --- | | s | | |  | | --- | | s | |
| o | |  | | --- | | "" | | |  | | --- | | "" | | o | o | o | o | so | so |
| n | “” | “” | o | on | on | on | so | so |
| e | “” | “” | o | on | on | one | one | one |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | l | o | n | g | e | s | t |  |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| s | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| t | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 2 |
| o | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 3 |
| n | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 4 |
| e | 0 | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 5 |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |

|  |  |
| --- | --- |
| I=5 | J=7 |
| 2 | 1 |

list1.stream()

Creates a stream of the list elements:

Stream<String> -> "apple", "orange"

.flatMapToInt(String::chars)

Each string is converted to its character code points (int values). So:

"apple" → 'a', 'p', 'p', 'l', 'e'

"orange" → 'o', 'r', 'a', 'n', 'g', 'e'

Resulting IntStream (all characters in order, flattened):

['a', 'p', 'p', 'l', 'e', 'o', 'r', 'a', 'n', 'g', 'e']

.mapToObj(c -> (char) c)

Converts each int back to a char object, resulting in:

['a', 'p', 'p', 'l', 'e', 'o', 'r', 'a', 'n', 'g', 'e']

.collect(Collectors.groupingBy(c -> c, LinkedHashMap::new, Collectors.counting()))

Groups by character and counts occurrences, using a LinkedHashMap to maintain insertion order.

Intermediate result:

{

'a' -> 2,

'p' -> 2,

'l' -> 1,

'e' -> 2,

'o' -> 1,

'r' -> 1,

'n' -> 1,

'g' -> 1

}

.entrySet().stream()

.map(e -> e.getKey() + "-" + e.getValue())

Converts each entry to a string of the form: char-count

Example: 'a'=2 becomes "a-2"

Intermediate list:

["a-2", "p-2", "l-1", "e-2", "o-1", "r-1", "n-1", "g-1"]

.collect(Collectors.joining(", "))

Joins the strings with comma + space:

"a-2, p-2, l-1, e-2, o-1, r-1, n-1, g-1"

Final Output

a-2, p-2, l-1, e-2, o-1, r-1, n-1, g-1