Worksheet: Finding maximal index of goal in matrix

Below is a code fragment intended to find the maximal (largest) index of a goal value that occurs multiple times in the input matrix. Could we make the code more efficient? If so, explain why?

```java
1. class AsyncFinishEurekaSearchMaxIndexOfGoal {
2.     HjEureka eurekaFactory() {
3.         comparator = (cur, newVal) -> { // cur is initially [-1, -1]
4.             (cur.x==newVal.x) ? (newVal.y - cur.y) : (newVal.x - cur.x)
5.         }
6.         return new MaximaEureka([-1, -1], comparator)
7.     }
8.     int[] doWork(matrix, goal) {
9.         val eu = eurekaFactory()
10.        finish (eu, () -> {
11.            forasyncChunked (0, matrix.length - 1, (r) ->
12.                procRow(matrix(r), r, goal));
13.        });
14.        return eu.get()
15.     }
16.     void procRow(array, r, goal) {
17.         for (int c = 0; c < array.length(); c++)
18.             if (goal.match(array(c)) offer([r, c]) // updates cur in eureka
19.     }
```

This code is inefficient due to starting c at 0 instead of array.length() -1. We could also use forasyncChunked to reduce the number of tasks created.