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?

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

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