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?

```
class AsyncFinishEurekaSearchMaxIndexOfGoal {
  HjEureka eurekaFactory() {
    comparator = (cur, newVal) -> { // cur is initially [-1, -1]
      (cur.x==newVal.x) ? (newVal.y - cur.y) : (newVal.x - cur.x) }
    return new MaximaEureka([-1, -1], comparator)
  }
  int[] doWork(matrix, goal) {
    val eu = eurekaFactory()
    finish (eu, () -> { // eureka registration
      forasync (0, matrix.length - 1, (r) ->
        procRow(matrix(r), r, goal));
    });
    return eu.get()
  }
  void procRow(array, r, goal) {
    for (int c = 0; c < array.length(); c++)
      check([r, c]) // terminate if comparator returns negative
      if goal.match(array(c)) offer([r, c]) // updates cur in eureka
  } }
```

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.