

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

