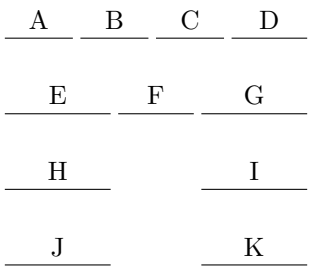
CS 1510 Homework 2

Brian Falkenstein, Brian Knotten, Brett Schreiber

August 28, 2018

1

Proof by counterexample:
Consider the following input:



The given Least Overlaps algorithm will clearly select F as the first interval as it has the least number of overlaps (2) and B and C will be eliminated as the overlapping intervals. Next, the algorithm will arbitrarily select another interval as each of the remaining intervals overlaps 3 other intervals. Assuming A is selected, E, H, and J will be eliminated and the algorithm will proceed with selecting D, eliminating G, I, and K and emptying S.
Therefore, the solution generated by the algorithm on the above input is A, F, D:

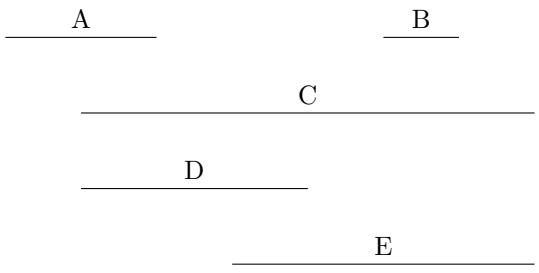
However, the obvious solution given the above input is A, B, C, D:

Therefore the algorithm does not solve the problem of generating a maximum cardinality subset of intervals given a set of intervals.

2

a

This algorithm A does not find the optimal solution for the following instance I: For the first room, $A(I)$ chooses the interval that ends earliest, A. This eliminates C and D as candidates for the first room. The next room that ends earliest is B, which eliminates E, and now there are no intervals left, so the first room is full. C, D, and E all overlap, so they must all have their own room. So $A(I)$ requires 4 rooms in its output: $\{A, B\}, \{C\}, \{D\}, \{E\}$. But a better solution exists using three rooms: $\{A, E\}, \{D, B\}, \{C\}$. So the algorithm A is not correct in all instances.



b

Assume the algorithm given, A, does not solve the interval coloring problem. Let s be the maximum number of intervals that overlap at any point, and thus, the optimal output (minimum number of rooms after assigning all intervals). Then, A will output a value larger than s for some input, and thus, A does not solve the problem. We can conclude that in the case that A's output is greater than s, this means that A assigned a class to a new room when another room already had a space for it, as this would be the only case where A's output could be greater than s. Because, in this algorithm, classes already placed are never altered / assigned to another room, this is a contradiction. The algorithm checks the set R of rooms that already have a class scheduled in it, and places the class in the first room it finds that does not cause an overlap. Thus, the output of A cannot be larger than s.