(3/4)

CS 1510
Algorithm Design

Greedy Algorithms

Problems 1 and 2

Due August 27, 2014

Buck Young and Rob Brown

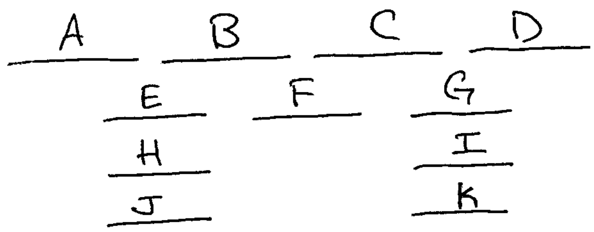
3/4

Problem 1

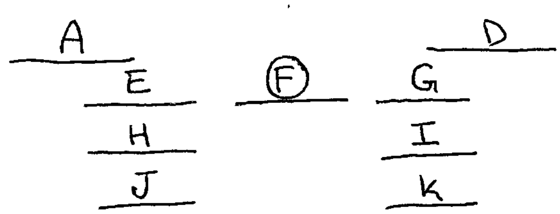
Theorem: That the given algorithm X is correct for all inputs in S.

Proof: This theorem will be proven incorrect by way of counter-example.

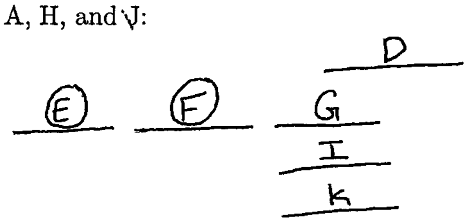
Consider these inputs:



Here we clearly see that the maximum cardinality subset of S is 4, given by the set {A, B, C, D}. This is optimal. Algorithm X, however, would choose the interval which overlaps the fewest number of other intervals first (choosing F). At this point, we throw out B and C:



At this point, we would choose any other interval (as they are all tied in overlaps) - let's choose E and toss out the overlaps A, H, and J:



Finally, we choose any other interval again (as they are all again tied in number of overlaps). Let's choose G and throw out D, I, and K:

Here we see that algorithm X has given us a subset with a cardinality of 3 (one less than optimal).

⇒ Thus, algorithm X is **not** correct for all inputs and our theorem is proven false.

Problem 2

(a) This algorithm does solve the Interval Coloring Problem.

0

Suppose we have a function G that returns $\{R_1, R_2, ..R_m\}$ where R_i is a set of intervals assigned to a given room according to the optimal algorithm discussed in class.

From this, we know that each room assignment R_i a) contains no overlapping intervals and b) has the maximum possible cardinality. Since each set has the maximum possible cardinality, and

$$|\bigcup_{i=1}^{n} R_i| = \sum_{i=1}^{n} |R_i|$$

is a mathematical fact, we know that the rooms as a whole have the maximum possible cardinality (ie, number of classes assigned). Since the maximum number of classes have been assigned to the current set of rooms, clearly the minimum number of rooms have been used. Not true

(b) This algorithm does solve the Interval Coloring Problem.

1

From a high-level, intuitive standpoint – the **only** justification for creating a new room is upon an overlap. This is an important interaction when we also consider the given hint (the maximum number of interval overlaps is equivalent to the lower bound for the number of rooms needed).

Let's dive a little deeper and consider any given interval I $\in$ S. The only instance where I is added to a **new** room is when it overlaps with an interval in **every** other room. Clearly, rooms are only added when absolutely necessary. So let's consider the very last room to be added. This room was created with an interval that must overlap an interval in every other room - its number of overlaps is equal to the total number of rooms. Thus, given the hint and this interaction, we can see that the algorithm is indeed optimal.