1. This algorithm does not solve the problem of finding a maximum cardinality set of non-overlapping intervals. Consider the following intervals:

\[ \text{A} \quad \text{B} \quad \text{C} \quad \text{D} \quad \text{E} \]

Obviously, the optimal solution is \{A, B, C, D\}. However, the interval that overlaps with the fewest others is E, and the algorithm will select E first, which precludes it from picking intervals B and C.

2. (a) This algorithm does not solve the interval-coloring problem. Consider the following intervals:

\[ \text{A} \\
\text{B} \quad \text{C} \quad \text{D} \\
\text{E} \quad \text{F} \quad \text{G} \]

The optimal solution is to put A in one room, \{B, C, D\} in another, and \{E, F, G\} in another, for a total of 3 rooms. However, maximizing the number of classes in the first room results in having \{B, C, F, G\} in one room, and classes A, D, and G each in their own rooms, for a total of 4.

(b) This algorithm does solve the interval-coloring problem. Note that if the greedy algorithm creates a new room for the current class \(c_i\), then because it examines classes in order of start times, \(c_i\)'s start point must intersect with the last class in all of the current rooms. Thus when greedy creates the last room, \(N\), it is because the start time of the current class intersects with \(N - 1\) other classes. But we know that for any single point in any class it can only intersect with at most \(s\) other class, it must be then that \(N \leq s\). As \(s\) is a lower bound on the total number needed and greedy is feasible it is thus also optimal.

4. (a) This greedy algorithm is optimal. We prove by contradiction. Assume greedy is not optimal for input \(I\), we pick the optimal solution, \(OPT\), that is identical to greedy for the most consecutive gas stations. Consider the first gas station where the greedy solution, \(G\), and \(OPT\) differ, call it station \(k\). Say \(G\) adds \(g_k\) gas and \(OPT\) adds \(o_k\) gas. We now create a new solution, \(OPT'\) as follows: \(OPT'\) is identical to \(OPT\) at every station except \(k\) and \(k + 1\). Call the amount of
gas $OPT$ adds at station $k+1$, $o_{k+1}$. At station $k$, $OPT'$ only adds $g_k$ gas to the tank, and at station $k+1$, $OPT'$ adds $o_{k+1} + (o_k - g_k)$. Clearly, $OPT'$ is identical to $G$ for one more station, namely $k$. We claim that $OPT'$ is feasible and spends no more time filling the tank than $OPT$. Prior to station $k+1$, $OPT'$ is identical to $G$ thus, because $G$ makes it to $k+1$, $OPT'$ must make it to $k+1$. By the fact that greedy adds the minimal amount of gas required to get from $k$ to $k+1$, and $G$ and $OPT$ differ at $k$, it must be that $o_k > g_k$, thus $o_{k+1} + (o_k - g_k) > 0$ meaning $OPT'$ adds a valid amount of gas at $k+1$. Further, because $g_k + (o_{k+1} + (o_k - g_k)) = o_k + o_{k+1}$, $OPT'$ has the same amount of gas in the tank as $OPT$ after filling up at $k+1$, namely $o_k + o_{k+1} - g_k$. Because $OPT'$ is identical to $OPT$ after $k+1$, $OPT'$ never runs out of gas after $k+1$. Finally, because the total gas put in the tank by $OPT'$, over $k$ and $k+1$, is $g_k + (o_{k+1} + (o_k - g_k)) = o_k + o_{k+1}$, $OPT$ and $OPT'$ add the same amount of gas in total over the two stations in which they differ, making their total time spent filling the same. Thus we have an optimal solution that is identical to greedy for one more station, a contradiction.

(b) This greedy algorithm is not optimal. Without loss of generality we can assume the car starts at $A$ with an empty tank. Consider the input of $x_1 = 0$, $x_2 = 5$, $x_3 = 6$, further, assume that $c$, $F$, and $r$ are such that a full tank of gas takes you 5km. The greedy algorithm will fill the tank twice but filling the tank only at $x_1$ then adding just enough at $x_2$ to go 1km will give a lower total time filling up.

6. (a) This algorithm is correct for the problem of minimizing the total sum of all line penalties. The proof is by contradiction. Assume there is an optimal solution $T$, and call the output of the greedy algorithm $G$. Let $s_i$ be the penalty of the $i$th line of solution $S$. Let $j$ be the number of the first line in $T$ that is different from the $j$th line in $G$. By the definition of the algorithm, $g_i < t_i$. Create a new solution $T'$ by moving the first word of line $i + 1$ in $T$ to the end of line $i$. Let $l$ be the length of this word. Note that $t'_{i+1} = t_{i+1} + l$ and $t'_i = t_i - l$. Therefore, the the total sum of all line penalties in $T'$ is the same as the total sum of all line penalties of $T$. $T'$ is more like greedy than $T$, and has the same total penalty. Contradiction.

(b) This algorithm is incorrect for the problem of minimizing the maximum line penalty. Let $L = 5$, and consider the words “AAA”, “BB”, “CC”, and “DDDD”. The greedy algorithm produces

<table>
<thead>
<tr>
<th>Word</th>
<th>Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>0</td>
</tr>
<tr>
<td>BBQ</td>
<td>3</td>
</tr>
<tr>
<td>DDDD</td>
<td>1</td>
</tr>
</tbody>
</table>

for a maximum line penalty of 3. The optimal solution is
AAA penalty = 2
BBCC penalty = 1
DDDD penalty = 1

for a maximum line penalty of 2.