The following scheduling algorithm is the one we studied in class.
(1) Sort the activities by increasing finishing time, $f_i$.
(2) Set $A = \{1\}$. Set $j = 1$.
(3) For $i = 2, \ldots, n$:
   If $s_i \geq f_j$:
       $A = A \cup \{i\}$.
       $j = i$.

The following algorithm was suggested to replace the one we studied in class.
(1) Sort the activities by increasing size of the interval $f_i - s_i$.
(2) Set $A = \{1\}$.
(3) For $i = 2, \ldots, n$:
   If activity $i$ does not conflict with any of the activities in $A$:
       $A = A \cup \{i\}$.

Consider the following set of three activities.

<table>
<thead>
<tr>
<th></th>
<th>$s_i$</th>
<th>$f_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

1. Which activities will the first algorithm select?
Answer: 1[0,5) and 3[5,10)

2. Which activities will the second algorithm select?
Answer: 2[4,6)

3. What can you say about the ability of the second algorithm to select the maximum number of activities?
Answer: It does not always select the maximum number of activities.