COMS 3101 - Fall 2013
Homework 3 (Extra)

- This part is optional.
- Due by start of class (Monday 4pm).
- See submission instructions.

1. Recall the game from HW#1, Extra:
   Choose any positive integer (1,2,…), call it X.
   If X is an even integer you divide X by 2.
   If X is an odd integer you multiply X by 3 and add 1.
   You continue this procedure until at some point you get 1 as your next integer. At that point you stop.

   a) Read the MATLAB documentation for the 'cell' array (help cell).
   b) Write a function named "ehw3b.m" which has one input argument {N}, and one output argument {A}.
      Your function should do the following: you are going to play the game in reverse order. So starting from 1 you will backtrack:

<table>
<tr><td rowspan="8">1</td><td rowspan="8">2</td><td rowspan="8">4</td><td rowspan="8">8</td><td rowspan="8">16</td><td rowspan="4">32</td><td rowspan="4">64</td><td rowspan="2">128</td><td rowspan="2">256</td><td>512</td></tr>
<tr><td>85</td></tr>
<tr><td>21</td><td>42</td><td>84</td></tr>
<tr><td rowspan="4">5</td><td rowspan="4">10</td><td rowspan="2">20</td><td rowspan="2">40</td><td>80</td></tr>
<tr><td>13</td></tr>
<tr><td>3</td><td>6</td><td>12</td></tr>
</table>

      Note: we ignore the cycle {1,2,4}.
      The input parameter N is the number of steps you are going to backtrack. For each step, you should store all the integers you reach in the corresponding element of your cell array A.
      For example, element 5 of A should be 16, element 6 of A should be the vector [5, 32].

c) Your function should produce the following plot:
   1. Your x axis should run from 1 to N.
   2. Your y axis from 0 to the 'maximum among all integers in the cell array A' + 1.
   3. For each element of the array A, you will plot red points, where x = index, y = integer. For example, for element 6 of A you should plot two points (6,5) and (6, 32).
d) The backtracking step above should be a separate sub-function. You can name this sub-function as you wish, but note:
   1. It is a sub-function, not a separate file.
   2. You make a call to your 'sub' to compute a particular element of A, or to compute the whole array, it is up to you.

2. Implement the "Selection Sort" algorithm in matlab. You can read about different sorting algorithms (selection sort among them) on Wikipedia or any other source (algorithms book). Of course MATLAB has the familiar built-in sort function that simply implements an efficient sort algorithm.

   a) Write a function named "ssort.m" which performs selection sort on a vector that is passed as an input argument (v). Your function should return the sorted vector (sv) as output. You should sort in "DESCENDING" order (largest element first). Keep in mind that some pseudo-code for selection sort is written assuming ascending order.
   b) Plot the sorted vector using the 'stem' function