Assignment. This is an individual lab.
For this part of the lab, develop a greedy algorithm for the Optimal Announcement Schedule problem (defined in the lecture notes distributed to you), and implement it in Python.

Create an `optimalAnnouncement.py` program that includes the `optimalAS()` function. The `optimalAS()` function shall take as input a list \( A = [(s_1, e_1), ..., (s_n, e_n)] \) of activities that are taking place. Here, \( s_i \) is the activity start time, and \( e_i \) is the activity end time (assume \( e_i > s_i \) for all \( i \)).

It shall return back a list \( \text{Times} = [t_1, ..., t_k] \) of times when the announcement needs to be made (or - in the terms in which the Optimal Announcement problem has been described in the handouts - the optimal stab set, i.e., the stab set with the smallest number of points).

In addition to the `optimalAS()`, your code shall include a simple test harness that generates a random instance of the Optimal Announcement problem, and runs `optimalAS()` on it, and outputs the results.

You can reuse the code from Part 1 of this Lab to set up the test harness, as the input to the Optimal Announcement Schedule problem and the input to the Activity Selection problem is essentially the same (it is just interpreted differently by the respective algorithms).

Deliverables. The `optimalAnnouncement.py` file. There is no submission until Lab 6 Part 3 which will concentrate on setting up proper testing harnesses for the two algorithms you implemented and running them.

Note. We will have discussed the Optimal Announcement problem in class, but the actual algorithm design for this problem is left up to you individually. There are several greedy strategies that could be discussed here. Your goal is to determine the correct greedy strategy and implement it correctly. You can, as an exercise, implement multiple greedy strategies and see if any of them can be empirically shown to be non-optimal. We will have some discussion of what the correct greedy strategy might be in class, but we won't be in position to prove it formally, so the final analysis of the greedy strategies is left up to you. Your final Lab 6 Part 3 report will contain proofs of optimality of your selected greedy strategy.