In this lab you will build a cottage implementation of the Activity Selection greedy algorithm. The pseudocode was discussed in class, but really, the entire algorithm construction process boils down to two things:

- Knowing the statement of the problem and the inputs and expected outputs
- Knowing the greedy criterion of extending the optimal solution

Problem Definition. Develop a function `SelectActivities(A)` which takes as input a NumPy array `A`. The array has `N` rows (for some integer number `N`) and 2 columns. Each row `j` contains two integer numbers `Start_j` and `End_j` such that `Start_j < End_j`). The function shall return a subset of rows of `A` that form a optimal set of selected activities.

Greedy Criterion for extending the optimal solution. Given a partial solution `S`, select on each step the activity *that ends the earliest* that has no conflict with any of the activities in `S`.

Solution process. As discussed in class, the algorithm starts by sorting the input list of activities in ascending order by the activity end time. Your Python solution *instead of sorting* the data using the `.sort()` method, or instead of writing your own sorting function implementation *will take advantage of your implementation of Priority Queues* from Lab 5. Notice that the Activity Selection procedure does not require a fully sorted array of activities - what it requires is the ability to retrieve the activities from the list in the ascending order of the end of activity time. Priority Queues work well for this - you can insert all activities into the priority queue at the beginning (using the end time as the sort key), and then use `ExtractMin()` to grab the next activity, and check whether that activity should be added to the solution.

Output. You have flexibility when it comes to producing output. The output of your function can be whatever is convenient for you: a list or array of `(Start_j, End_j)` pairs, or a list or array of indexes of the selected activities in the array `A`, or a boolean array, where an element is set to True if the activity under that index is selected.