1. In class we discussed interval packing problems. Here we explore interval covers.

(a) You are given a set of \( n \) intervals \([s_i, f_i]\) and a range \([0, T]\). You would like to find a minimal set \( I \subseteq [n] \) of intervals whose union covers the range. That is, we say that \( I \) is a valid cover if, for all \( t \in [0, T] \), there exists an \( i \in I \) such that \( t \in [s_i, f_i] \). Give (and prove correctness for) a greedy algorithm to compute a valid cover with the smallest number of intervals, in linear time after sorting.

(b) (Optional) Now suppose that each interval also has a cost \( c_i \), and your goal is to find a valid cover \( I \) minimizing the total cost \( \sum_{i \in I} c_i \). Give a dynamic programming solution to this problem that takes \( O(n^2) \), or even \( O(n \log n) \), time.

2. There’s a Jupyter Notebook linked from the class webpage.