Let $I_1, \ldots, I_n$ denote the $n$ intervals. We say that an $I_j$-restricted solution is one that contains the interval $I_j$.

Here is an algorithm, for fixed $j$, to compute an $I_j$-restricted solution of maximum size. Let $x$ be a point contained in $I_j$. First delete $I_j$ and all intervals that overlap it. The remaining intervals do not contain the point $x$, so we can “cut” the time-line at $x$ and produce an instance of the Interval Scheduling Problem from class. We solve this in $O(n)$ time, assuming that the intervals are ordered by ending time.

Now, the algorithm for the full problem is to compute an $I_j$-restricted solution of maximum size for each $j = 1, \ldots, n$. This takes a total time of $O(n^2)$. We then pick the largest of these solutions, and claim that it is an optimal solution. To see this, consider the optimal solution to the full problem, consisting of a set of intervals $S$. Since $n > 0$, there is some interval $I_j \in S$; but then $S$ is an optimal $I_j$-restricted solution, and so our algorithm will produce a solution at least as large as $S$.  

\footnote{ex434.357.684}