Solution
The algorithm to solve this problem is as follows:
While there remains an interval n (represented as a pair of start and stop times in set S) that has gone unexamined, grab one and look it at its start time. Compare this start time to every other interval $i \in S$. If s_n lies within another interval $\in S$ such that $t_i > s_n > s_i$, then add one to a count keeping track of how many intervals in S the interval n intersects with. Once n has looked at every other interval, record the count of how many total intervals n intersected with. If this is the first n that has been examined, set this count to be the maxCount. Otherwise, if this is not the first n to be examined, update the maxCount only if n's count is higher than the previous maxCount. Once every $n \in S$ has been examined, return the maxCount.

Each iteration of this algorithm involves comparing each element in the set to every other element in the set. Therefore this will take $O(n^2)$ time.

This algorithm requires a data structure to hold all n pairs, so it will take O(n) space.