

Analysis: Parenting Partnering Returns

Test Set 1

We can solve this test set by naively trying every possible subset of activities to be covered by Jamie and assign the rest of the activities to be covered by Cameron. For each subset of activities, we can check whether a pair of activities overlap for each pair of activities. An activity with start time s_1 and end time t_1 overlaps with another activity with start time s_2 and end time t_2 if the time intersection is not empty (i.e., $\max(s_1, s_2) < \min(t_1, t_2)$).

The running time of this solution is $O(2^N \times N^2)$, which is fast enough to solve Test Set 1.

Test Set 2

We can solve this test set by greedily assigning the activities in increasing order of start time. For each activity (in increasing order of start time), we can check whether Jamie or Cameron can be assigned to cover the activity and assign the activity to whomever can be assigned to (or arbitrarily if both partners can be assigned). The check can be done by iterating all activities that have been previously assigned to Jamie and Cameron.

The greedy assignment is correct because the only way that the assignment fails is when there is a time that is covered by three activities. In such a case, there is indeed no valid assignment. When deciding who to assign an activity with start time s , only activities with start times no later than s have been assigned. Therefore, if both Jamie and Cameron have some activity assigned with end time later than s , it means that there are three activities that use the time between s and $s + 1$, and therefore, there is no possible assignment. If an assignment is possible, there cannot be any set of three activities that pairwise overlap, so by the contrapositive of the previous argument, we will be able to assign the activity to at least one of Jamie or Cameron at every step.

The running time of this solution is $O(N^2)$, which is fast enough to solve this test set. To optimize the solution to $O(N \log N)$ time, we can efficiently check whether an activity can be assigned to Jamie or Cameron by keeping track of the end time of the last activity assigned to each partner and comparing this to the start time of the new activity. In this case, only $O(N)$ extra time is needed after sorting the activities by their start time.

Graph approach

Another possible approach to solve this test set is to construct a [graph](#) with N nodes, each representing one activity. We add an edge connecting a pair of nodes if the pair of activities represented by the nodes overlap (see Test Set 1 section for details on how to check if two intervals overlap). This graph is commonly known as an [interval graph](#).

Therefore, the problem is equivalent to [finding a partition of nodes C and J such that every edge connects a node in C and a node in J](#), as we can assign all activities represented by nodes in C to Cameron and all activities represented by nodes in J to Jamie. The running time of the algorithm to find the partition (or report if one does not exist) is linear on the size of the graph. The graph has N nodes and $O(N^2)$ edges, which means the solution requires $O(N^2)$ time to build the graph and $O(N^2)$ time to run the partition algorithm, so also $O(N^2)$ time overall.