1.
    a. If a movie that was very long ended last then it would not work
    b. Again, if a movie was very long it would interfere with other movies that could make it so that you could watch more movies
    c. This always constructs an optimal schedule because it leaves the same amount of movies still available when it recurs
    d. If the shortest movie interferes with two other movies that don't interfere with each other then this algorithm would select one movie as optimal not 2
    e. If one movie conflicts with three movies total a movie at the beginning, end, and one in the middle that it's longer than. The movie at the end conflicts with some as well. The algorithm would pick the one in the middle and delete the one that spans it even though it could be part of the optimal solution
    f. If a movie with the longest duration conflicted with another movie that also conflicted with a third movie it would delete the longest one and keep only one of the others
    g. This would work because it would always leave the optimal amount of movies remaining
    h. This wouldn't work because a movie that ends last might be really long and also start first meaning that it gets selected when other movies don't overlap
2. If n is odd add an increment one to the end of the steps. Then subtract one from n. If n is even add a times 2 to the end of the steps. It's greedy because at any moment possible we reduce the number n by the most that it can be reduced by. This is proven by subtracting 1 from any odd number n or dividing 2 by any even number n. The same algorithm will work just as well. Also after doing that operation we're left with the result after using the greedy step for the first step
3. Find the smallest value in R and the largest value in L that is smaller than that. Find any value that is between those two values and add it to an array. It works because once it has found one working element it's in the same situation it was in before.