1. First show that this problem has optimal substructure, i.e. show that the optimal solution to the problem for the building of height n is an extension of the optimal solution for a shorter building.

Assume set S is an optimal solution to the problem containing a sequence of balconies, and assume there is a highest balcony hi included in this optimal solution of set S. Then S’ is the solution to the problem with height hi and same set of balconies. Assume S’ is not optimal, and there is some better solution S’’ that uses fewer number of balconies. However, adding balcony hi to the optimal solution for the subproblem S’’ would be the optimal solution and S would not be the optimal solution, which is a contradiction.

Thus, we prove this problem has optimal substructure by contradiction.

1. Next give the greedy choice function for this problem, i.e. give a way to identify some balcony guaranteed to be used in some optimal solution. Show the correctness of your greedy choice function using an exchange argument.

The correct greedy choice function for this problem is to always select the lowest balcony that Jackie can safely jump to, i.e., the balcony that is at most k meters lower than Jackie's current height and is closest to the ground.

To show the correctness of this greedy choice function, we will use an exchange argument. Let's assume that our greedy choice function does not produce an optimal solution, and let O be the optimal solution. This means that there exists some balcony in O that is not included in the sequence produced by our greedy choice function. Assume this balcony is the first choice in the sequence O called i. This balcony i has to be higher than the first choice included in our greedy choices called j. If the height that Jackie jumps from is n = 2k and there is a balcony m at height k, our greedy choices j will be this balcony m, and the greedy choice i from sequence O must be higher than j. In this case, sequence O would have at least 2 balconies and our greedy choice would have only 1 balcony. Sequence would not be the optimal solution, which is a contradiction.

1. Finally, Use the previous two items to devise a greedy algorithm to solve this problem. Analyze your algorithm’s running time.

First, initialize the sequence list and current height variable equal to n.

Secondly, under a while loop with condition current height greater than k, looping through the balcony list from lowest to highest. If the balcony i has a distance to n less than or equal to k, jump to this balcony i (updating the current height to i).

Finally, return the sequence.

The running time of this algorithm would be O(m^2) since we need to iterate through potentially all balconies for each level of current height, and all levels of current heights is a subset of all balconies.