Maximum Achievable Profit of Thief

1. **Define the entries of your table in words. E.g. T(i) or T(i, j) is ...**
   T(i) is the maximum total profit the thief can achieve when he burglarizes some subset of i consecutive houses in a neighborhood, where the house is labeled 1, 2, …, i.

2. **State a recurrence for the entries of your table in terms of smaller subproblems. Don't forget your base case(s).**
   Base case:
   for i = 0, T(i) = 0
   for i = 1, T(i) = p(1)

   Recurrence:
   for 2<=i<=n: T(i) = max{T(i-2) + p[i] , T(i-1)}

3. **Write pseudocode for your algorithm to solve this problem.**
   when i = 0:
   T(i) = 0
   when i = 1:
   T(i) = p[1]
   for i = 2 to n:
   T(i) = max{T(i-2)+p[i], T(i-1)}
   return T(n)

4. **State and analyze the running time of your algorithm.**
   when i = 0: O(1)
   when i = 1: O(1)
   for i = 2 to n: O(n-1)
   return T(n): O(1)

   The running time can be represented by: 3*O(1) + O(n-1) ➔ O(n)