1473: Paint House III

Input:

$\boxed{houses}$: [1, 2, 2, 3, 3, 2, 1, 1] len = m

5 neighborhoods = [{1}, {2, 2}, {3, 3}, {2}, {1, 1}]

n = no. of colors

Each number in the houses list refer to the color by which house can be colored.

$\boxed{cost}$ = m x n matrix

cost[i][j] → represents (house at index i) cost of coloring that house with color j+1.

color values $1 \le c \le n$

∴ 0 index for j in cost will represent color 1.

$\boxed{target}$ → The number of neighborhoods we can have.

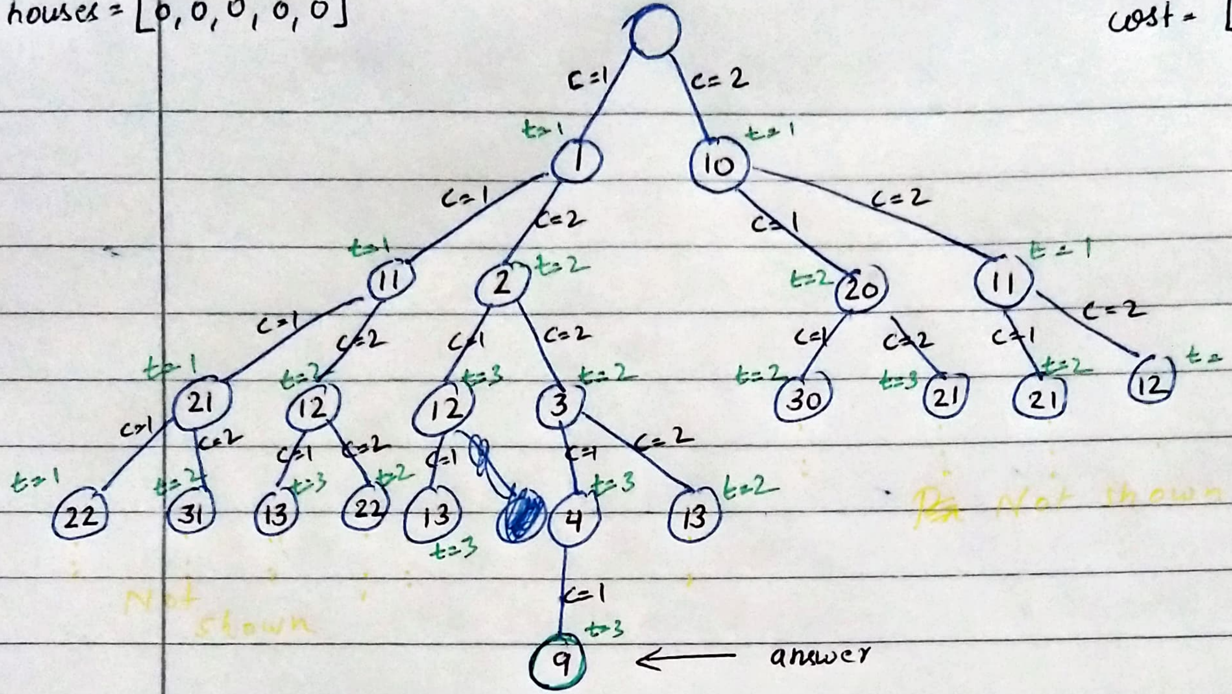
⇒ Return the minimum cost of painting remaining houses in such a way that there are exactly "target" neighborhoods. Return -1 if not possible.

We can only color those houses that are not already colored.

houses = [0, 0, 0, 0, 0]

Here in first example all houses have color 0 (i.e not colored). if any house has color (already) ~~from~~ with values [1 ~~s~~, n], then we have skip them (Keeping target in mind) houses = [0, 0, 0, 0, 0]

cost = [[1, 10], [10, 1], [10, 1], [1, 10], [5, 1]]



In above tree, we have not shown the complete tree, since they were not going to provide the mimom answer,

- In above case we had only two colors, (the number of children each node can have is K (where k is total colors)

- We increment the t (current target) value if current color and previous color don't match. You can see in above tree when the value of c is changed t is increased by 1.

- Once t == target, we keep applying the same color until we reach the end.

- Skip the house that is already colored, increment the t if house is of different color from previous one.

- All leaves for which target != t return INT_MAX, as they may contain lower values (smaller) than the desired answer.

- This is a 3 dimensional DP problem. Check all 3 solutions.

  $T - O(m \cdot n^2 \cdot target)$