

5431. Paint House III

[My Submissions \(/contest/weekly-contest-192/problems/paint-house-iii/submissions/\)](/contest/weekly-contest-192/problems/paint-house-iii/submissions/)

[Back to Contest \(/contest/weekly-contest-192/\)](/contest/weekly-contest-192/)

There is a row of m houses in a small city, each house must be painted with one of the n colors (labeled from 1 to n), some houses that has been painted last summer should not be painted again.

A neighborhood is a maximal group of continuous houses that are painted with the same color. (For example: houses = [1,2,2,3,3,2,1,1] contains 5 neighborhoods [{1}, {2,2}, {3,3}, {2}, {1,1}]).

Given an array `houses`, an $m * n$ matrix `cost` and an integer `target` where:

- `houses[i]` : is the color of the house i , **0** if the house is not painted yet.
- `cost[i][j]` : is the cost of paint the house i with the color $j+1$.

Return the minimum cost of painting all the remaining houses in such a way that there are exactly `target` neighborhoods, if not possible return **-1**.

User Accepted:	0
User Tried:	2
Total Accepted:	0
Total Submissions:	2
Difficulty:	Hard

Example 1:

Input: `houses = [0,0,0,0,0]`, `cost = [[1,10],[10,1],[10,1],[1,10],[5,1]]`, $m = 5$, $n = 2$, `target = 3`
Output: 9

Explanation: Paint houses of this way [1,2,2,1,1]
 This array contains `target = 3` neighborhoods, [{1}, {2,2}, {1,1}].
 Cost of paint all houses $(1 + 1 + 1 + 1 + 5) = 9$.

Example 2:

Input: `houses = [0,2,1,2,0]`, `cost = [[1,10],[10,1],[10,1],[1,10],[5,1]]`, $m = 5$, $n = 2$, `target = 3`
Output: 11
Explanation: Some houses are already painted, Paint the houses of this way [2,2,1,2,2]
 This array contains `target = 3` neighborhoods, [{2,2}, {1}, {2,2}].
 Cost of paint the first and last house $(10 + 1) = 11$.

Example 3:

Input: `houses = [0,0,0,0,0]`, `cost = [[1,10],[10,1],[1,10],[10,1],[1,10]]`, $m = 5$, $n = 2$, `target = 1`
Output: 5

Example 4:

Input: houses = [3,1,2,3], cost = [[1,1,1],[1,1,1],[1,1,1],[1,1,1]], m = 4, n = 3, target

Output: -1

Explanation: Houses are already painted with a total of 4 neighborhoods [{3},{1},{2},{3}]

Constraints:

- $m == \text{houses.length} == \text{cost.length}$
- $n == \text{cost}[i].\text{length}$
- $1 \leq m \leq 100$
- $1 \leq n \leq 20$
- $1 \leq \text{target} \leq m$
- $0 \leq \text{houses}[i] \leq n$
- $1 \leq \text{cost}[i][j] \leq 10^4$

JavaScript



```
1  /**
2   * @param {number[]} houses
3   * @param {number[][]} cost
4   * @param {number} m
5   * @param {number} n
6   * @param {number} target
7   * @return {number}
8   */
9  var minCost = function(houses, cost, m, n, target) {
10
11  };
```

☐ Custom Testcase

Use Example Testcases