Problem 3332: Maximum Points Tourist Can Earn

Problem Information

Difficulty: Medium
Acceptance Rate: 47.10%
Paid Only: No
Tags: Array, Dynamic Programming, Matrix

Problem Description

You are given two integers, `n` and `k`, along with two 2D integer arrays, `stayScore` and `travelScore`.

A tourist is visiting a country with `n` cities, where each city is **directly** connected to every other city. The tourist's journey consists of **exactly** `k` **0-indexed** days, and they can choose **any** city as their starting point.

Each day, the tourist has two choices:

* **Stay in the current city** : If the tourist stays in their current city `curr` during day `i`, they will earn `stayScore[i][curr]` points. * **Move to another city** : If the tourist moves from their current city `curr` to city `dest`, they will earn `travelScore[curr][dest]` points.

Return the **maximum** possible points the tourist can earn.

Example 1:

Input: n = 2, k = 1, stayScore = [[2,3]], travelScore = [[0,2],[1,0]]

Output: 3

Explanation:

The tourist earns the maximum number of points by starting in city 1 and staying in that city.

Example 2:

Input: n = 3, k = 2, stayScore = [[3,4,2],[2,1,2]], travelScore = [[0,2,1],[2,0,4],[3,2,0]]

Output: 8

Explanation:

The tourist earns the maximum number of points by starting in city 1, staying in that city on day 0, and traveling to city 2 on day 1.

Constraints:

* `1 <= n <= 200` * `1 <= k <= 200` * `n == travelScore.length == travelScore[i].length == stayScore[i].length` * `k == stayScore.length` * `1 <= stayScore[i][j] <= 100` * `0 <= travelScore[i][j] <= 100` * `travelScore[i][i] == 0`

Code Snippets

C++:

```
class Solution {
public:
int maxScore(int n, int k, vector<vector<int>>& stayScore,
vector<vector<int>>& travelScore) {

}
};
```

Java:

```
class Solution {
public int maxScore(int n, int k, int[][] stayScore, int[][] travelScore) {

}
}
```

Python3:

```
class Solution:
def maxScore(self, n: int, k: int, stayScore: List[List[int]], travelScore:
List[List[int]]) -> int:
```