

1947. Maximum Compatibility Score Sum

There is a survey that consists of n questions where each question's answer is either 0 (no) or 1 (yes).

The survey was given to m students numbered from 0 to $m - 1$ and m mentors numbered from 0 to $m - 1$. The answers of the students are represented by a 2D integer array `students` where `students[i]` is an integer array that contains the answers of the i th student (**0-indexed**). The answers of the mentors are represented by a 2D integer array `mentors` where `mentors[j]` is an integer array that contains the answers of the j th mentor (**0-indexed**).

Each student will be assigned to **one** mentor, and each mentor will have **one** student assigned to them. The **compatibility score** of a student-mentor pair is the number of answers that are the same for both the student and the mentor.

- For example, if the student's answers were `[1, 0, 1]` and the mentor's answers were `[0, 0, 1]`, then their compatibility score is 2 because only the second and the third answers are the same.

You are tasked with finding the optimal student-mentor pairings to **maximize** the **sum of the compatibility scores**.

Given `students` and `mentors`, return *the maximum compatibility score sum that can be achieved*.

Example 1:

```
Input: students = [[1,1,0],[1,0,1],[0,0,1]], mentors = [[1,0,0],[0,0,1],[1,1,0]]
```

```
Output: 8
```

Explanation: We assign students to mentors in the following way:

- student 0 to mentor 2 with a compatibility score of 3.
- student 1 to mentor 0 with a compatibility score of 2.
- student 2 to mentor 1 with a compatibility score of 3.

The compatibility score sum is $3 + 2 + 3 = 8$.

Example 2:

```
Input: students = [[0,0],[0,0],[0,0]], mentors = [[1,1],[1,1],[1,1]]
```

```
Output: 0
```

Explanation: The compatibility score of any student-mentor pair is 0.

```

def maxCompatibilitySum(self, students: List[List[int]], mentors:
List[List[int]]) -> int:
    res = [-1]
    visited = [False]*len(students)
    #ssf ==> score so far
    #res ==> final result. It is a list as it acts as a global
variable
    #idx ==> index over students array.
    self.maxCompatibilityUtil(students,mentors,0,visited,res,0)
    return res[0]

def maxCompatibilityUtil(self,students,mentors,ssf,visited,res,idx):
    if idx==len(students):
        res[0] = max(res[0],ssf)
        return

    for i in range(len(mentors)):
        if visited[i]==False:
            visited[i]=True
            temp = 0
            for j in range(len(mentors[i])):
                if mentors[i][j]==students[idx][j]:
                    temp = temp+1

    self.maxCompatibilityUtil(students,mentors,ssf+temp,visited,res,idx+1)
    visited[i]=False

```