**★★★☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：12875: Concert Tour**

**解題者：蔡明軒**

**解題日期：2022年4月28日**

**題意：**

題目有多筆測資，每組測資給定商店的數量、演唱會的數量，每場演唱會在每個商店所帶來的利潤，以及商店移動到任意商店的成本 (注意沒有移動也可能會有成本，第一場演唱會到任何商店都不用成本)，求最大的總利潤

**題意範例：**

3 4 // 三個商店 四場演唱會

1 3 20 40

50 20 1 2

20 50 50 1

0 10 10

10 0 10

一張含有 桌 的圖片

自動產生的描述一張含有 桌 的圖片

自動產生的描述10 10 0

**解法：**

用一個二維陣列紀錄狀態， dp[i][j] = n ，代表第 j 個演唱會在第 i 個商店的狀態下最高收益為 n，若j = 1，這狀態的最高利潤就是第 i 個商店的第一場的收益，否則計算前一天在哪個商店扣掉成本後所能得到收益最大，最後得到 dp[1~商店總數][演唱會總數] 的最大值即為答案。

DP 轉移式 (k=1~商店總數):

一張含有 文字 的圖片

自動產生的描述**解法範例：**

計算dp[1][2](第2場演唱會辦在S1的最大利潤):

k=1: 從 S1 移動到 S1 的成本為 0 ，1-0=1

k=2: 從 S2 移動到 S1 的成本為 10 ，50-10=40

k=3: 從 S3 移動到 S1 的成本為 10 ，20-10=10

前一場演唱會辦在 S2 的效益最好，

一張含有 桌 的圖片

自動產生的描述dp[1][2] = 40 + profit[1][2] = 40+3 = 43

一張含有 桌 的圖片

自動產生的描述計算完所有的 dp 表格後，找出最後一場的淨利最大值，這個值即是答案，以上面的的測資為例，答案為 170。

**討論：**

(1) 解動態規劃的題目時，記得要把結果存起來，遇到已經算過的位置時，要直接回傳之前算過的結果來。

(2) 利用memset 函式可以幫助我們快速初始化陣列。

一張含有 文字, 牆, 銀色 的圖片

自動產生的描述memset(dp,0,sizeof(dp));

**程式：**

#include <bits/stdc++.h>

#define pi acos(-1)

#define endl '\n'

#define MaxN 1005

#define EPS 1e-6

typedef long long ll;

using namespace std;

int s, c;

int profit[101][51], cost[101][101];

int dp[101][51];

int calc(int now, int d) {

    if (dp[now][d]) return dp[now][d];

    if (d == 1) return profit[now][1];

    int max = calc(1, d - 1) - cost[1][now];

    for (int i = 2; i <= s; i++) {

        int val = calc(i, d - 1) - cost[i][now];

        if (val > max) max = val;

    }

    return dp[now][d] = (max + profit[now][d]);

}

int main() {

    ios::sync\_with\_stdio(0);

    cin.tie(0);

    int n;

    cin >> n;

    while (n--) {

        memset(profit, 0, sizeof(profit));

        memset(cost, 0, sizeof(cost));

        memset(dp, 0, sizeof(dp));

        cin >> s >> c;

        for (int i = 1; i <= s; i++)

            for (int j = 1; j <= c; j++) cin >> profit[i][j];

        for (int i = 1; i <= s; i++)

            for (int j = 1; j <= s; j++) cin >> cost[i][j];

        int ans = calc(1, c);

        for (int i = 2; i <= s; i++) {

            int val = calc(i, c);

            if (val > ans) ans = val;

        }

        cout << ans << '\n';

    }

    return 0;

}