Maximum path sum that starting with any cell of 0-th row and ending with any cell of (N-1)-th row

> 给定NxN的矩阵matrix，求从第一行任意元素开始到达最后一行任意元素的所有路径中，和最大的路径，并返回其路径和；在移动过程中，对于元素(i, j)，其下一步只能往(i + 1, j - 1)、(i + 1, j)或者(i + 1, j + 1)三个位置之一移动。

采用自下而上的动态规划，注意边界

```
#include <bits/stdc++.h>

using namespace std;

typedef vector<int> vi;
typedef vector<vector<int>> vvi;

int maxPathSum(vvi &matrix, int &N) {
    vvi dp(N, vi(N));
    for (int j = 0; j < N; ++j) dp[N - 1][j] = matrix[N - 1][j];
    for (int i = N - 2; i >= 0; --i) {
        for (int j = 0; j < N; ++j) {
            if (j == 0)
                dp[i][j] = max(dp[i + 1][j], dp[i + 1][j + 1]) + matrix[i][j];
            else if (j == N - 1)
                dp[i][j] = max(dp[i + 1][j - 1], dp[i + 1][j]) + matrix[i][j];
            else
                dp[i][j] = max(max(dp[i + 1][j - 1], dp[i + 1][j]), dp[i + 1][j + 1]) + matrix[i][j];
        }
    }
    return *max_element(dp[0].begin(), dp[0].end());
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        vvi matrix(N, vi(N));
        for (int i = 0; i < N; ++i)
            for (int j = 0; j < N; ++j)
                scanf("%d", &matrix[i][j]);
        printf("%d\n", maxPathSum(matrix, N));
    }
}
```