Maximum path sum in a triangle

> 数塔问题，从第一层开始，每次只能选择下一层的相邻元素，求第一层到最后一层的最大和

dp[i][j]表示到第i层第j个元素的最大和，自下而上递推，状态转移方程如下：

$$dp[i][j] = max(dp[i+1][j], dp[i+1][j+1]) + triangle[i][j];$$

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

using namespace std;

int maxPathSum(int &N, vector<vector<int>> &triangle) {
    vector<vector<int>> dp(N, vector<int>(N));
    for (int i = 0; i < N; ++i) {
        dp[N- 1][i] = triangle[N - 1][i];
    }
    for (int i = N - 2; i >= 0; --i) {
        for (int j = 0; j <= i; ++j) {
            dp[i][j] = max(dp[i + 1][j], dp[i + 1][j + 1]) + triangle[i][j];
        }
    }
    return dp[0][0];
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        vector<vector<int>> triangle(N, vector<int>(N));
        for (int i = 0; i < N; ++i) {
            for (int j = 0; j <= i; ++j) {
                scanf("%d", &triangle[i][j]);
            }
        }
        printf("%d\n", maxPathSum(N, triangle));
    }
    return 0;

}
```