Adjacents are not allowed

> 给定一个2 * N的数组，从任意点开始，每次选择一个数，要求所选择的数与上一次选择的数不能相邻（水平、垂直或对角），求所能够选择到的所有数字的最大和

初始在第一列时，维护两个变量：includeColSum——表示选择当前列中的数之后的最大和，excludeColSum——不选择当前列中的数的最大和，初始化includeColSum = max(arr[0][0], arr[1][0])，excludeColSum = 0；

从第二列开始（索引下标从1开始），maxExcludeCurColSum记录不包含当前列的最大值，然后更新includeColSum与excludeColSum

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

using namespace std;

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

int maxSum(vvi &arr, int &N) {
    int includeColSum = max(arr[0][0], arr[1][0]), excludeColSum = 0;
    for (int col = 1; col < N; ++col) {
        int maxExcludeCurColSum = max(includeColSum, excludeColSum);
        includeColSum = excludeColSum + max(arr[0][col], arr[1][col]);
        excludeColSum = maxExcludeCurColSum;
    }
    return max(includeColSum, excludeColSum);
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N;
        scanf("%d", &N);
        vvi arr(2, vi(N));
        for (int i = 0; i < 2; ++i)
            for (int j = 0; j < N; ++j)
                scanf("%d", &arr[i][j]);
        printf("%d\n", maxSum(arr, N));
    }
}
```