Maximum Sum More Optimised:

int kadane2D(int array[N+1][M+1]){

// Modify the array's elements to now hold the sum

// of all the numbers that are above that element in its column

for (int i = 1; i <= N; i++) {

for (int j = 1; j <= M; j++){

array[i][j] += array[i-1][j];

}

}

int ans = 0; // Holds the maximum sum matrix found till now

for(int top=1; top<=N; top++){

for(int bottom=top; bottom<=N; bottom++){

// loop over all the N^2 sub problems

int[] sums = new int[N+1];

// store the sum of numbers between the two rows

// in the sums array

for(int i=0; i<=N; i++){

sums[i] = array[bottom][i] - array[top-1][i];

}

// O(n) time to run 1D kadane's on this sums array

ans = Math.max(ans, kadane1d(sums));

}

}

return ans;

}