363. Max Sum of Rectangle No Larger Than K

题目描述：https://leetcode.com/problems/max-sum-of-sub-matrix-no-larger-than-k/

> 在一个二维矩阵中找到一个矩形，这个矩形的元素加和不大于k，且是所有不大于k的矩形中加和最大的一个。
> 例如：
>
> ```
> Example:
> Given matrix = [
>   [1,  0, 1],
>   [0, -2, 3]
> ]
> k = 2
> The answer is 2. Because the sum of rectangle [[0, 1], [-2, 3]] is 2 and 2 is the max number no larger than k (k = 2).
> ```

解题思路：

类似于max of vector no larger than k
就是先把sum[i]代表从0-i的和，
然后找lower_bound(sum-k)
然后结果就是 sum-*it 这个就拓展到二维

代码：

```
class Solution {
public:
    int maxSumSubmatrix(vector<vector<int>>& matrix, int k) {
        if(matrix.size() == 0) return 0;
        int m = matrix.size(), n = matrix[0].size();
        int res = INT_MIN;
        for(int i = 0; i < n; i++) {
            vector<int> sum(m, 0);
            for(int j = i; j < n; j++) {
                for(int l = 0; l < m; l++) {
                    sum[l] += matrix[l][j];
                }
                set<int> sums;
                sums.insert(0);
                int maxSum = INT_MIN, rsum = 0;
                for(int l = 0; l < m; l++) {
                    rsum += sum[l];
                    auto it = sums.lower_bound(rsum-k);
                    if(it != sums.end()) {
                        maxSum = max(maxSum, rsum-*it);
                    }
                    sums.insert(rsum);
                }
                res = max(res, maxSum);
            }
        }
        return res;
    }
};
```