bool searchMatrix(int\*\* matrix, int matrixSize, int\* matrixColSize, int target){

    int mid;

    int right;

    int left;

    for(int i = 0; i< matrixSize ;i++){//將每一個row打開

        mid = matrixColSize[i]/2;

        right = matrixColSize[i]-1;

        left=0;

        //printf("left[%d]=%d mid[%d]=%d right[%d]=%d\n",left,matrix[i][left],mid,matrix[i][mid],right,matrix[i][right]);

        while(1){

            //printf("left=%d mid=%d right=%d\n",matrix[i][left],matrix[i][mid],matrix[i][right]);

            if(matrix[i][mid] == target){//如果mid找到就return答案

                //printf("get Ans %d",nums[mid]);

                return 1;

            }else if(matrix[i][right] == target){//如果right找到就return答案

                //printf("get Ans %d",nums[right]);

                return 1;

            }else if(matrix[i][left] == target){//如果left找到就return答案

                //printf("get Ans %d",nums[left]);

                return 1;

            }else if(matrix[i][mid] > target){//target比mid的值小

                right=mid;//就往左側找

                mid=(right+left)/2;

            }else{//target比mid的值大

                left=mid;//就往右側找

                mid=(right+left)/2;

            }

            if( right== mid || left== mid){//如果找到答左側或右側都找不到mid==right或mid==left就跳出迴圈

                printf("break");

                break;

            }

        }

    }

return false;

}