CS 3310 HW #2

1.
```
int[] find (int[] arr, int left, int right, int min, max, int L, int R) {
    int minResult;
    if (left + 1 >= right)
        then minResult = Math.min (a[left], min)
            if (a[left] - min > max) {
                then max = a[left] - min
                     R = left
            }
    } else {
        int mid = (right - left) / 2
        minResult = find (arr, left, mid-1, min, max)
        minResult = find (arr, mid, right, minResult, max)
    }
    return minResult;
```

2.
```
int turningPoint (int[] arr, int left, int right) {
    int center = (right - low) / 2
    if (arr[center] > arr[center-1] && arr[center] > arr[center+1])
        then return center
    else if (arr[center] < arr[center+1])
        then turningPoint (arr, center+1, right)
    else if (arr[center] <= arr[center-1])
        then turningPoint (arr, left, center)
}
```

Best-Case:	Wast-Case:	Average Case
$\Omega(1)$	$T(n) = T(n/2) + c$	$T(n) = T(n/2) + c$
	$= O(\log n)$	$= \Theta(\log n)$

```
}. or) int maxDiff (int[] arr) {
    Arrays.sort(arr); int sum1=0; int sum2=0;
    for (i=0; i<arr.length/2-1; ++) {
        sum1 += arr[i]
    }
    for (i=arr.length/2; i<arr.length; i++) {
        sum2 += arr[i]
    }
    return sum2 - sum1;
```

b) `int maxDiff(int[] arr) {`

```
    int sum=0; sum=0;
    for (int i=0; i<arr.length; i++) {
        sum+ = arr[i]
    }
    int avg = sum / arr.length
    int lowSum=0, highSum=0
    for (int i=0; i<arr.length; ++) {
        if (arr[i] ≥ avg)
            highSum += arr[i]
        else
            lowSum += arr[i]    }
    return highSum - lowSum;
```

4.

```
boolean searchMatrix(int[][] matrix, int target) {
    int rowLow = 0, colLow = 0, rowHigh = length-1, colHigh = matrix[0].length;
    while rowLow < rowHigh {
        int center = (rowHigh + rowLow)/2
        if (matrix[center][colLow] == target) {
            return true
        }
        if (matrix[center][colLow] > target) {
            rowHigh = center -1
        } else if (matrix[center+1][colLow] > target) {
            rowHigh = center
            rowLow = center
        } else {
            rowLow = center+1
        }
    }
    while (colLow <= colHigh) {
        int mid = (colHigh + colLow)/2
        if (matrix[rowLow][mid] == target) {
            return true
        }
        if (matrix[rowLow][mid] > target) {
            colHigh = mid-1
        } else {
            colLow = mid+1
        }
    }
    return false
}
```

5.	2	5	7	12		1	0	2	7
	4	1	4	2		5	9	3	4
	5	6	3	8	x	6	1	2	8
	1	6	7	9		1	2	8	5

~~$m1 = (2+1+3+9)(1+9+2+5)$~~

$m2 =$

$m1 = (2+1)(1+9) = 30$

$m2 = (4+1)(1) = 5$

$m3 = 2(0-9) = -18$

$m4 = 1(5-1) = 4$

$m5 = (2+5)(9) = 63$

$m6 = (4-2)(1+0) = 2$

$m7 = (5-1)(5+9) = 56$

$m1 = (7+2)(2+4) = 63$

$m2 = (4+2)(2) = 12$

$m3 = 7(7-4) = 21$

$m4 = 2(3-2) = 2$

$m5 = (7+12)4 = 76$

$m6 = (4-7)(2+7) = -42$

$m7 = (12-2)(3+4) = 70$

$m1 = m(5+6)(6+2) = 88$

$m2 = (1+6)(6) = 42$

$m3 = 5(1-2) = -5$

$m4 = 6(1-6) = -30$

$m5 = (5+6)2 = 22$

$m6 = (1-5)(6+1) = -28$

$m7 = (6-6)(1+2) = 0$

$m1 = (2+5)(3+8) = 84$

$m2 = (7+9)(2) = 32$

$m3 = 3(8-5) = 9$

$m4 = 9(8-2) = 54$

$m5 = (3+8)(5) = 55$

$m6 = (7-3)(2+8) = 64$

$m7 = (8-5)(8+5) = -13$

$30+4-63+56$ $\quad -18+63$ $\quad 63+2-76+70$ $\quad 21+76$

$5+4$ $\quad 30-18+5+2$ $\quad 12+2$ $\quad 63+21+12-42$

$88-30-22+0$ $\quad -5+22$ $\quad 84+54-55+-13$ $\quad 9+55$

$42-30$ $\quad 88-5+42-28$ $\quad 32+54$ $\quad 84+9+32+64$

27 $\quad$ 45 $\quad$ 59 $\quad$ 97

9 $\quad$ 9 $\quad$ 14 $\quad$ 30

5.

2	5	7	12
4	1	4	2
5	6	3	8
1	6	7	9

$\times$

1	0	2	7
5	9	3	4
6	1	2	8
1	2	8	5

$$\begin{bmatrix} 2 & 5 \\ 4 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 5 & 9 \end{bmatrix}$$

$m_1 = (2+1)(1+9) = 30$
$m_2 = (4+1)(1) = 5$
$m_3 = 2(0-9) = -18$
$m_4 = 1(5-1) = 4$
$m_5 = (2+5)(9) = 63$
$m_6 = (4-2)(1) = 2$

$$\begin{bmatrix} 27 & 45 & 59 & 97 \\ 9 & 9 & 14 & 30 \\ 36 & 17 & 70 & 64 \\ 12 & 43 & 86 & 125 \end{bmatrix}$$