Name: Shuo Wu Student ID:452666 wustl key: wushuo

Q3.

a.

Step 1: we can divide the n x n matrix into four (*n/2* x *n/2*) smaller matrices. For example, the center element **9** partitions the matrix into four matrices as shown in the picture below. Since the four smaller matrices are also sorted both row and column-wise, the problem can naturally be divided into four sub-problems.

Step 2: Assume the target number is 18, which is greater than the center element 9, we can eliminate the upper left matrix since all element in this matrix is smaller than it rightest-lowest element which is 9. We also can assume our target number is 8, which is smaller than 9. Similarly, we can drop the bottom right matrix for future calculation since the elements in this matrix are all bigger than 9. Please note however, we still need to search the upper right and bottom left quadrant, even though the example below seems to show all elements in the two mentioned quadrants are greater than 9.

Step3: Of course, if the center element is our target element, we have found the target and stop searching. If not, we proceed by searching the rest of three quadrants.

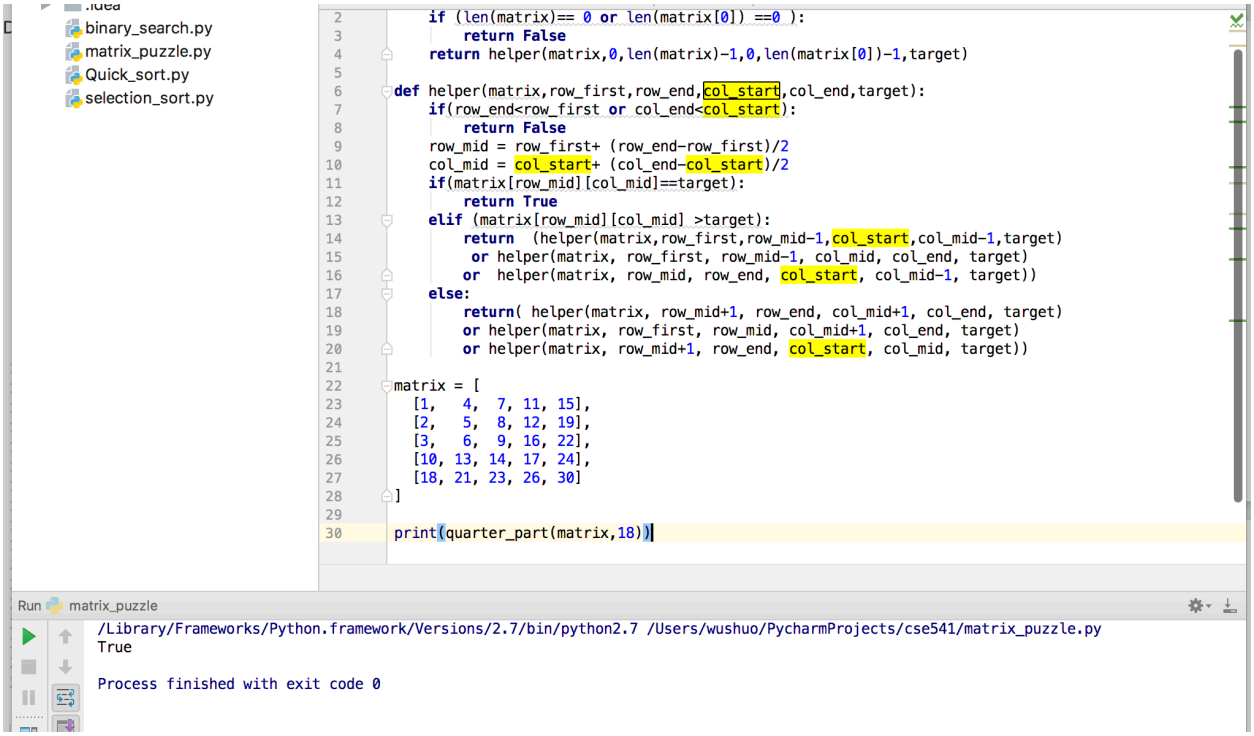
1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

pseudo-code:

```
def quarter_part( matrix, target):
    if (len(matrix)== 0 or len(matrix[0]) ==0 ):
        return False
    return helper(matrix,0,len(matrix)-1,0,len(matrix[0])-1,target)

def helper(matrix,row_first,row_end,col_start,col_end,target):
    if(row_end<row_first or col_end<col_start):
        return False
    row_mid = row_first+ (row_end-row_first)/2
    col_mid = col_start+ (col_end-col_start)/2
    if(matrix[row_mid][col_mid]==target):
        return True
    elif (matrix[row_mid][col_mid] >target):
        return  (helper(matrix,row_first,row_mid-1,col_start,col_mid-1,target)
         or helper(matrix, row_first, row_mid-1, col_mid, col_end, target)
        or  helper(matrix, row_mid, row_end, col_start, col_mid-1, target))
```

```
    else:
        return( helper(matrix, row_mid+1, row_end, col_mid+1, col_end, target)
        or helper(matrix, row_first, row_mid, col_mid+1, col_end, target)
        or helper(matrix, row_mid+1, row_end, col_start, col_mid, target))
```



B,
$T(n) = 3T(n/2) + c$,
where n is the dimension of the matrix.

C,
Using master method,
We know that
$T(n) = 3T(n/2) + c$,
$= 3 [3T(n/4) + c] + c$
$= 3 [3 [3T(n/8) + c] + c] + c$
$= 3^k T(n/2^k) + c (3^k - 1)/2$
$= 3^k (T(n/2^k) + c) - c/2$

Setting $k = \lg n$,
$T(n) = 3^{\lg n} (T(1) + c) - c/2$
$= O(3^{\lg n})$
$= O(n^{\lg 3})$ <== $3^{\lg n} = n^{\lg 3}$
$= O(n^{1.58})$