SHUBHAM GUPTA
180749

Theo - Ass 1

3.) pseudocode:

```
q3(A, B, n) {
    if (n==1) {
        return (A[0] + B[0]) /2
    }
    if (n==2) {
        return (max(A[0], B[0]) + min(A[1], B[1]))/2
    m1 <- median(A, n)
    m2 <- median(B, n)
    if (m1 == m2) {
        return m1;
    } else if (m1 < m2) {
        if (n % 2 == 0) {
            return q3(A + n/2 - 1, B, n - n/2 + 1);
        }
        return q3(A + n/2, B, n - n/2);
    } else if (m1 > m2) {
        if (n % 2 == 0) {
            return q3(B + n/2 - 1, A, n - n/2 + 1)
        }
        return q3(B + n/2, A, n - n/2);
    }
}

median(A, n) {
    if (n % 2 == 0) {
        return (A[n/2] + A[n/2 + 1]) /2
    } else {
        return (A[n/2]);
    }
}
```

algorithm:

we first calculate median of both arrays lets name them m_A & m_B, ~~[illegible]~~ now we have 3 cases.

C_I = $m_A = m_B$ then m_A is the median.

C_{II} = $m_A > m_B$ then median is present b/w
↳ $A[0] - A[m/2]$
↳ $B[m/2] - B[m-1]$

C_{III} = $m_A < m_B$ then median is present b/w
↳ $A[m/2] - A[m-1]$
↳ $B[0] - B[m/2]$

we repeat this process until the search size becomes 2 or 1

for 2 we output $[\max(A[0], B[0]) + \min(A[1], B[1])]/2$

for 1 we output $(A[0] + B[0])/2$.

time complexity : $O(\log m)$

space complexity : $O(1)$

Proof of correctness:

C_1 $m = 1$ we can clearly do $(A[0] + B[0])/2$

C_2 $m = 2$ we —"— $[\max(A[0], B[0]) + \min(A[1], B[1])]/2$

C_3 $m_1 > m_2$, ~~[illegible]~~

lets merge A & B into D

$D[0]$ — $B[mid]$ — $A[mid]$ — $D[2m-1]$

it can be seen that median lies b/w $A[mid]$ & $B[mid]$ ∴ if we take a subarray either $D[0]$ to $A[mid]$ or $B[mid]$ to $D[2m-1]$, we can get the median.

C_4 $m_1 < m_2$ can be proved similarly in above fashion