Zijian Gao
26319560
2/29/2016
CS311 HW2

9.3-8.)

The median m is defined to be the number at which there are n/2 numbers in a given set greater than m, and n/2 numbers less than m.

For X[1..n] and Y[1..n], let m_1 be the median index for X and m_2 be the median index for Y, these can be calculated in constant time depending on whether n is even or odd.

If $m_1 = m_2$, the median has already been found as there must be an equal (n/2) elements greater than/less than both m_1 and m_2 in X and Y respectively, so when joined together, there would be n elements greater than this value and n elements less than this value, so choose either m_1 or m_2 to be the median value. This is also the base case for recursion.

If $m_1 < m_2$, $Y[\lceil \frac{n}{2} \rceil .. n]$ can be disregarded as any numbers greater than m_2, are trivially greater than the median of the joined array. In the same manner, $X[1..\lfloor \frac{n}{2} \rfloor]$ can also be disregarded as the first n/2 elements of X must be less than the median of the joined array. So, the two sub-arrays that are of concern are X'=$X[\lceil \frac{n}{2} \rceil .. n]$ and Y'=$Y[1..\lfloor \frac{n}{2} \rfloor]$. Now, recursively find the medians for X' and Y', each time reducing the amount of elements of concern by a factor of 2 for each subarray, which in total reduces the total number of elements by a factor of 2. $T(n) = 2T(\frac{n}{2}) + \theta(1)$ and so the runtime is $O(\log n)$ by the master method.

If $m_1 > m_2$, the opposite of the above applies, where $Y[1..\lfloor \frac{n}{2} \rfloor]$ and $X[\lceil \frac{n}{2} \rceil .. n]$ subarrays can be eliminated from consideration and the same recursive steps may be applied to reduce the resulting sub-array into half-sized arrays until the base case ($m_1 == m_2$) is reached