EECS 336 Fall 2015
Homework Problem 4.2

Find median of twin arrays A and B each of size $n > 1$ with distinct values across both arrays. Both arrays are sorted from lowest to highest. Without losing generality, we assume length of A, B is even in the algorithm shown below.

The idea is to compare the median of A, B and continually split each of A and B in halves until each sub-array is of size 1 or the difference between medians of the sub-arrays is 1. The sub-array is analogous to binary search algorithm, where half of the sub-array that cannot possibly contain true median of $A \bigcup B$ is discarded.

Algorithm 1 Calculate median in twin arrays

procedure MEDIAN(A, B)
    if $|A| = 1$ and $|B| = 1$ **then**
        return $max(A, B)$
    else
        Find median of arrays A, B:
        if n is even **then**
            $m_A = A[\frac{n}{2} + 1]$
            $m_B = B[\frac{n}{2} + 1]$
        else
            $m_A = A[\lceil \frac{n}{2} \rceil]$
            $m_B = B[\lceil \frac{n}{2} \rceil]$
        end if
        Split A, B in halfs in order:
        if n is even **then**
            $A_1 = \{A[0], A[1], ..., A[\frac{n}{2}]\}$
            $A_2 = \{A[\frac{n}{2} + 1], ..., A[n]\}$
            $B_1 = \{B[0], B[1], ..., B[\frac{n}{2}]\}$
            $B_2 = \{B[\frac{n}{2} + 1], ..., B[n]\}$
        else
            $A_1 = \{A[0], A[1], ..., A[\lceil \frac{n}{2} \rceil]\}$
            $A_2 = \{A[\lceil \frac{n}{2} \rceil], A[\lceil \frac{n}{2} \rceil + 1], ..., A[n]\}$
            $B_1 = \{B[0], B[1], ..., B[\lceil \frac{n}{2} \rceil]\}$
            $B_2 = \{B[\lceil \frac{n}{2} \rceil], B[\lceil \frac{n}{2} \rceil + 1], ..., B[n]\}$
        end if
        if $|m_A - m_B| \leq 1$ **then**
            return $max(m_A, m_B)$
        else
            if $m_A > m_B$ **then**
                return MEDIAN(A_1, B_2)
            else
                return MEDIAN(A_2, B_1)
            end if
        end if
    end if
end procedure

Runtime Analysis: Algorithm terminates when size of array in argument of MEDIAN decreases from n to 1. Thus, a total of $\log(n)$ steps are carried out. Each steps involves splitting the sub-arrays, which takes $O(1)$. Thus, the total runtime is $O(\log(n))$.

Correctness: By induction.

Base case: When $|A| = 1$ and $|B| = 1$, $max(A, B)$ is returned as the larger element in A, B combined.

Inductive hypothesis: For each call of MEDIAN(A', B'), true median of original twin arrays are in current sub-arrays A'_1, B'_2 or A'_2, B'_1 depending on values of m'_A, m'_B.

Inductive case: For each subsequent call of MEDIAN, true median of original twin arrays A, B is still one of the output sub-arrays.

Suppose after current call of MEDIAN, $m'_A < m'_B$. Then, since A', B' are split in halves according to ascending order of value, true median of twin arrays must be in the upper half of A' and lower half of B'. Thus, A'_2, B'_1 will be put into the recursive algorithm to find the next subset of arrays that contains true median. The recursive algorithm will be called until either (1) each sub-array is of size 1, and the larger value will be taken as the median, or (2) since all values in A, B are distinct, the fact that m'_A, m'_B differs by 1 indicates that these two values are the middle two elements in the combination of the twin arrays. Thus, $max(m'_A, m'_B)$ will yield the correct true median.