Problem 5

a) Since the size of the array is even, we can easily separate it into two sides. We apply binary search on both sides, since both sides are well ordered by the information we have. Therefore, the binary search will work.

   If both binary search returns "not found", then the integer not in the array.

   The time complexity of this algorithm should be the sum of two binary search of $\frac{n}{2}$.

   Therefore, runtime should be $\Theta\left(2 \times \log\frac{n}{2}\right) = \Theta\left(\log\frac{n^2}{4}\right) = \Theta(\log n^2)$

b) This time, we do something that similar to the selection sort. We set two pointers initially points to the beginning of the array and the end of the array respectively. We do 13 iterations, compare the two elements that pointers point to, and move the pointer which has the smaller element 1 index towards middle. By doing this, in the 13th iteration, the smaller elements that the pointers point to is the 13th smallest integer.

   Since we just iterate 13 times, and each iteration cost constant time.

   The runtime should be $\Theta(13) = \Theta(1)$