Assignment 2

An array $A[1, \ldots, n]$ is said to have a *majority element* if **more than half** of its entries are the same. Given an array, the task is to design an efficient algorithm to tell whether the array has a majority element, and, if so, to find that element. The elements of the array are not necessarily from some ordered domain like the integers, and so there can be no comparisons of the form “is $A[i] > A[j]?''$. (Think of the array elements as GIF files, say.) However you can answer questions of the form: “is $A[i] = A[j]?''$ in constant time.

1. Show how to solve this problem in $O(n \log n)$ time.
   (*Hint*: Split the array A into two arrays A_1 and A_2 of half the size. Does knowing the majority elements of A_1 and A_2 help you figure out the majority element of A? If so, you can use a divide-and-conquer approach.)
2. Can you give a linear-time algorithm?

   (*Hint*: Here's another divide-and-conquer approach:

   2.1 Pair up the elements of A arbitrarily, to get $n/2$ pairs

   2.2 Look at each pair: if the two elements are different, discard both of them; if they are the same, keep just one of them

   Show that after this procedure there are at most $n/2$ elements left, and that they have a majority element if and only if A does.)