Homework 3

1. You are given an array of n objects. You are told that some element occurs at least \([n/2] + 1\) times in the array. We call this element the majority element. Suppose the objects are totally ordered (that is you are given a function that takes two objects A and B as input and returns \((\text{key}(A) < \text{key}(B))\)). Give an algorithm that finds the majority element in time \(\Theta(n)\).

2. Give a tight asymptotic bound on the behavior of \(T\), defined by the recursion relation \(T(n) = T(n-1) + \ln(n)\) and \(T(1) = 1\).

3. Prove that \(\Omega(n \log n)\) is the tight lower bound for comparison based sorting algorithm with \(n\) input values.

4. Given an \(n\) by \(n\) matrix \(A\) with integers entries. Find four entries \(A[i_1,j_1], A[i_1,j_2], A[i_2,j_1], \text{ and } A[i_2,j_2] \) with \(i_1 < i_2\) and \(j_1 < j_2\) so that the sum \(A[i_1,j_1] + A[i_1,j_2] + A[i_2,j_1] + A[i_2,j_2]\) is maximal among such sums. Your algorithm should beat the \(O(n^4)\) bound.