McGill University COMP251: Assignment 1

Worth 10%. Due September 17 at the beginning of lecture (10am)

The work you submit must be your own. You may discuss problems with each others; however, you should prepare written solutions alone. Copying assignments is a serious academic offence, and will be dealt with accordingly.

Question 1 (a) Give a divide-and-conquer algorithm for the following problem:

- **Input**: Two arrays $A[1 \dots n]$ and $B[1 \dots n]$ of n, each array contains n elements which are sorted in non-decreasing order.
- **Output**: The median (i.e., the n-th smallest element) of the $2n$ elements in the arrays.

For example, given input $A = (1, 5, 6)$ and $B = (2, 4, 9)$, the output is 4.

We are interested in minimizing the total number of comparisons between elements of the arrays. (Imagine the situation where the elements are big objects so that the cost of comparing any two of them is significant.) So your algorithm must perform $\mathcal{O}(\ln n)$ number of comparisons between the elements. Explain why it indeed meets this requirement.

(b) Show that your algorithm works correctly.

Question 2 Consider the following algorithm (call ZZZ here) for sorting an array $A[1 \dots n]$ into non-decreasing order:

ZZZ-sort(A, i, j):

1. if $A[i] > A[j]$ do
2. swap $A[i]$ and $A[j]$
3. if $i + 1 \geq j$ return
4. $k \leftarrow \lfloor (j - i + 1)/3 \rfloor$ % round down one third of the length of $A[i \dots j]$
5. ZZZ-sort$(A, i + k, j)$ % call recursively on the last two-thirds
6. ZZZ-sort$(A, i, j - k)$ % call recursively on the first two-thirds
7. ZZZ-sort$(A, i + k, j)$ % call recursively on the last two-thirds again

(a) Prove that ZZZ-sort$(A, 1, n)$ correctly sorts the input array $A[1 \dots n]$.

(b) Give a recurrence for the worst-case running time of ZZZ-sort on an array of length n, and use this to give a tight asymptotic (i.e., using Θ-notation) bound on the worst-case running time.