COMP 182: Algorithmic Thinking
18 February 2014

1. Given a *sorted* array of *distinct* integers $A[0 \ldots n-1]$, we'd like to find whether there is an index i, $0 \leq i \leq n-1$, for which $A[i] = i$. Give the pseudo-code of a divide-and-conquer algorithm that runs in $O(\log n)$ time.

2. Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ be two sets of integers, whose intersection ($A \cap B$) we're interested in.

   (a) Give the pseudo-code of a brute-force algorithm for this problem, and determine its running time (using big-O notation).

   (b) Give the pseudo-code of an $O(n \log n)$ algorithm for this problem.

The Master Theorem Consider the recurrence

$$T(n) = aT(n/b) + f(n).$$

If $f(n) = \Theta(n^d)$ where $d \geq 0$, then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if} \quad a < b^d \\ \Theta(n^d \log n) & \text{if} \quad a = b^d \\ \Theta(n^{\log_b a}) & \text{if} \quad a > b^d \end{cases}.$$

Analogous results hold for the O and Ω notations as well.