Assignment 1

1 (Exercise 4.2-3) Draw the recursion tree for $T(n) = 4T(\lfloor n/2 \rfloor) + cn$, where c is a constant, and provide a tight asymptotic bound on its solution. Verify your bound by the substitute method.

2 Let $A[1:n]$ be an array of n distinct integers sorted in increasing order. (Assume, for simplicity, that n is a power of 2.) Give an $O(\log n)$-time algorithm to decide if there is an integer i, $1 \leq i \leq n$, such that $A[i] = i$. Your answer must include (a) a brief description of the main ideas (from which the correctness of the method should be evident), (b) pseudocode, and (c) an analysis of the running time.

3 (Exercise 6.2-3) What is the effect of calling $\textsc{Max-Heapify}(A, i)$ when the element $A[i]$ is larger than its children?

4 Given two sorted sequences with m, n distinct integers, respectively, design and analyze an efficient divide-and-conquer algorithm to find the kth smallest integer in the merge of the two sequences. The best algorithm runs in time $O(\log(\max(m, n)))$.

5 (Exercise 7.2-3) Show that the running time of QUICHSORT is $\Theta(n^2)$ when the array A contains distinct elements and is sorted in decreasing order.

6 Let S be a set of n points, $p_i = (x_i, y_i), 1 \leq i \leq n$, in the plane. A point $p_j \in S$ is a *maximal point of* S if there is no other point $p_k \in S$ such that $x_k \geq x_j$ and $y_k \geq y_j$. The figure below illustrates the maximal points of a point-set S. Note that the maximal points form a "staircase" which descends rightwards.

```
  ----o
      |
*     ----o
          |
    *     |             o = maximal point
          --o           * = non-maximal point
*           |
       *    ---o
               |
```

Give an efficient divide–and–conquer algorithm to determine the maximal points of S.

Your answer should include (i) a clear description (in words) of the main ideas and the data structures used, which makes the correctness self-evident, (ii) pseudocode for the algorithm, and (iii) an analysis of the running time and space.

7 (Exercise 8.3-4) Show how to sort n integers in the range 0 to $n^2 - 1$ in $O(n)$ time.

8 Let A be an $n \times n$ matrix of integers such that each row is strictly increasing from left to right and each column is strictly increasing from top to bottom. Given an $O(n)$-time algorithm for finding whether a given number x is an element of A, i.e., whether $x = A(i, j)$ for some i, j.