Datastructures and Algorithms

Assignment 1

Deadline: October 18, at 1.45pm

This deadline is hard as nails! Late submissions will not receive a grade.

Preferably hand in your solutions at the lecture; else send them to w.j.fokkink@vu.nl (in a single file).

Explain your answers, and provide your pseudocode descriptions with comments to explain the ideas behind your code.

1. Analyze the time complexity of the following two programs, in terms of $O$.

   (a) \textbf{Algorithm Loop1}(n):
       \begin{verbatim}
       s ← 0
       for i ← 1 to $n^2$ do
           for j ← 1 to i do
               s ← s + j
       \end{verbatim}
       \hspace{1cm} (6 pts)

   (b) \textbf{Algorithm Loop2}(n):
       \begin{verbatim}
       s ← 0
       for i ← 1 to n do
           for j ← 1 to n do
               for k ← 1 to i + j do
                   s ← s + i + j
       \end{verbatim}
       \hspace{1cm} (10 pts)
2. Let $insert(L, n)$ insert natural number $n$ into singly-linked list $L$.

Provide a pseudocode description for $insert(L, n)$ such that if $L$ is a sorted list and each natural number occurs at most once in $L$, then these two properties also hold for $insert(L, n)$. (So if $n$ already appears in $L$, then $n$ is not inserted into $L$.)

If $n$ is inserted in $L$ (i.e., if $n$ is not in $L$), then return $true$, and else $false$.

Also take into account (and if needed update) $first$ and $last$. (20 pts)

3. Sort the sequence

\[34 \ 12 \ 11 \ 18 \ 34 \ 9 \ 7 \ 15 \ 2.\]

using

(a) selection sort \hfill (6 pts)
(b) mergesort \hfill (6 pts)
(c) quicksort, always taking the first element of a (sub)list as pivot \hfill (10 pts)
(d) heapsort \hfill (10 pts)

In each case, show step by step how the sequence is sorted.

4. (a) Construct an AVL tree from the sequence

\[3 \ 5 \ 6 \ 1 \ 2 \ 4\]

by successively inserting the numbers in the list, starting from the empty tree. After each insertion, restructure the tree into an AVL tree if needed. \hfill (10 pts)

(b) Remove the root from the resulting AVL tree, and restructure the resulting binary search tree into an AVL tree again. (Explain how this AVL tree is constructed.) \hfill (6 pts)

5. We consider a hash table of length 11, and the hash function $h(k) = k \mod 11$. Add (in the given order) the numbers

\[1 \ 13 \ 2 \ 24 \ 10 \ 12 \ 4\]

to an initially empty hash table, where collisions are resolved using

(a) chaining \hfill (6 pts)
(b) open addressing with double hashing, using $h'(k) = 6 - (k \mod 5)$ \hfill (10 pts)

Explain in both cases how the place of each number in the hash table is computed.