Midterm Exam Data Structures and Algorithms 2014-2015

Friday September 26, 2014, 15.30-17.00

7 exercises

Answers may be given in English or in Dutch.

Exercise 1. *(8+7 points)*
This exercise is concerned with $\mathcal{O}$.

(a) Determine for the following pairs of functions f_i and g_i whether $f_i \in \mathcal{O}(g_i)$ and/or $g_i \in \mathcal{O}(f_i)$. You do not have to motivate your answer.

(i) $f_1(n) = \sum_{i=1}^{n} i$ and $g_1(n) = n^2 + 3n - 7$,
(ii) $f_2(n) = 5n \log n + n$ and $g_2(n) = 10n + 5$,
(iii) $f_3(n) = 3^n$ and $g_3(n) = 4^n$,
(iv) $f_4(n) = n^3$ and $g_4(n) = n^4$.

(b) What is the worst-case time complexity of the following program in terms of $\mathcal{O}$? Motivate your answer.

```
Algorithm Loop(n):
    p := 1
    for i := 1 to n^2 do
        for j = 1 to i do
            p := p + i
```

Exercise 2 *(7+8 points)*
This exercise is concerned with insertion sort.
The pseudo-code for insertion sort is given:

```
Algorithm insertionSort(A, n):
    for j := 2 to n do
        key := A[j]
        i := j − 1
        while i ≥ 1 and A[i] > key do
            A[i + 1] := A[i]
            i := i − 1
        A[i + 1] := key
```

(a) Apply insertion sort to the array $[4, 2, 3, 5, 1]$.

Give (at least) the intermediate result after every iteration of the for-loop.

(b) Explain why the worst-case time complexity of insertion sort is in $\mathcal{O}(n^2)$.

Exercise 3. (*8 points*)
This exercise is concerned with singly linked lists.

(a) For a singly linked list we have an operation first that gives the first node, and an operation last that gives the last node. In a node, we have operations element and next with the suggested meaning.

Give pseudo-code for the algorithm insertLast that takes as input a singly linked list L and data d, and that updates L by inserting at the end a new node with data (element) d.

Exercise 4. (*7+7+7 points*)
This exercise is concerned with quicksort.
The pseudo-code for partition is given:

```
Algorithm partition(A, p, r):
    x := A[r]
    i := p − 1
    for j = p to r − 1 do
        if A[j] ≤ x then
            i := i + 1
            exchange A[i] with A[j]
    exchange A[i + 1] with A[r]
    return i + 1
```

(a) Apply partition to the array $[4, 1, 5, 2, 3]$.

Give (at least) the result for every iteration of the for-loop.

(b) Give pseudo-code for the algorithm quicksort that takes as input an array and two indices in that array. You may use partition.

(c) Solve (step by step) the recurrence equation for quicksort:

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n-1) + n & \text{if } n > 1 \end{cases}$$

What is hence the worst-case time complexity of quicksort in terms of $\mathcal{O}$?

Exercise 5. (*8 points*)
This exercise is concerned with hashing.

(a) We consider a hash table of length 11 (an array with indices $1 \ldots 11$), and the hash function $h(k) = k \mod 11$. Add the following numbers in this order to the initially empty hash table:

1 13 2 24 10 12

solving collisions by open addressing with linear probing.

Exercise 6. (*8 points*)
This exercise is concerned with tree traversals.

(a) We consider binary trees implemented in a linked structure. The operation root gives the root of the tree. In a node, we have operations parent pointing to the parent, left pointing to the left child, and right pointing to the right child. (A pointer may be null.)

Give an algorithm (not necessarily in pseudo-code) that takes as input a binary tree T and a node v in that tree, and gives as output the node that is visited after v in a preorder traversal.

Exercise 7. (*7+8 points*)
This exercise is concerned with binary search trees and AVL-trees.

(a) Give the worst-case time complexity of searching in a binary search tree; motivate your answer.

(b) Construct an AVL-tree by inserting the numbers

3 5 6 1 2 4

one by one, starting from the empty tree. After each insertion, rebalance the tree if needed. Give your answer in pictures (with comments if needed).

The mark for the midterm is (the total number of points plus 10) divided by 10.