Exercises for Algorithms and Data Structures

Carsten.Schneider@risc.jku.at

1. Explain and discuss the notions "function", "algorithm" and "program" and describe the connections between them. Why does one differ between the different concepts?

2. Explain and discuss the notions "abstract data type", "data type", "data structure" and "class/module/package". Why does one differ between the different concepts?

3. Complete the data type

   algebra queue

   sorts queue, elem, bool

   ops

empty :		$\to$	elem
front :	queue	$\to$	elem
enqueue :	queue $\times$ elem	$\to$	queue
dequeue :	queue	$\to$	queue
isemty :	queue	$\to$	bool

   (a) Explain in "words" and with "pictures" how this data type can be implemented with arrays.

   (b) Which problem can arise after several executions of queue and dequeue? How can this problem be handled?

4. We are given the following data type for sets:

   algebra set

   sorts set, elem, bool

   ops

empty :		$\to$	set
insert :	set $\times$ elem	$\to$	set
delete :	set $\times$ elem	$\to$	set
member :	set $\times$ elem	$\to$	bool
isempty :	set	$\to$	bool
union :	set $\times$ set	$\to$	set
intersection :	set $\times$ set	$\to$	set
difference :	set $\times$ set	$\to$	set

   sets elem is a set; set= $\{S \subseteq \text{elem} \mid S \text{ is finite}\}$.

   functions ...

(a) Explain how this data type for set $= \{a_1, \ldots, a_m\}$ can be implemented with a bit vector. Indicate the worst case complexity of your algorithms.

(b) Describe the basic idea how the function union can be implemented with linked lists.

5. Find a closed form for the sequence $S(n)$ with $n \in \mathbb{N}$ that is determined by the recurrence

$$n\, S(n) - (n+2)\, S(n-1) = n, \quad n > 0$$

with the initial value $S(0) = 0$.

6. Present an algorithm that computes the minimum of the elements stored in a binary search tree.

   (a) Prove termination and correctness of your algorithm based on the properties of a binary search tree.

   (b) $P(n) = \sum_{k=1}^{n} \frac{1}{k} - \frac{2n}{n+1}$ is the average case length of a path from the root to a vertex of a binary search tree with n vertices. State a reason why the average case complexity of your algorithm that finds the minimum is $\mathcal{O}(\log(n))$.

7. What is the extra condition of a binary search tree in order to obtain an AVL tree? Justify why insert, delete, max and min have the worst case complexity $\mathcal{O}(\log n)$. Why cannot this complexity be observed for an ordinary binary search tree?

8. Describe the basic idea of *MergeSort* and *QuickSort*. Compare both algorithms.

9. Describe how polynomials in the polynomial ring $\mathbb{Z}[x]$ can be implemented with linked lists or hash tables.

10. Present two different possibilities how directed graphs can be represented in the memory. What are the advantages/disadvantages of both representations?

11. Present an algorithm that performs a depth-first traversal of a directed graph. How can this algorithms be used to decide if a graph has a cycle?