Reminders  Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Exercises  These should not be handed in, but the material they cover may appear on exams.

- Hash tables: CLRS Chapter 11.1. See also exercises 11.2-2 and 11.2-5.
- “Handshake Lemma”: Prove that in any undirected graph, the sum of the degrees of all vertices is even.

Problems to be handed in. *Please submit each problem on a separate sheet of paper.*

1. CLRS, Problem 12-2 (Radix Trees)

2. (Postfix) Consider the following syntax for specifying simple mathematical calculations: an *expression* can either be a number or an algebraic operation $\text{arg}_1 \text{arg}_2 \text{op}$, where $\text{op} \in \{+, -, \times, /\}$, and $\text{arg}_1, \text{arg}_2$ are *expressions*. The meaning of such an expressions is that operator $\text{op}$ should be applied to the outcomes of expressions $\text{arg}_1$ and $\text{arg}_2$. For example, $"1 2 + 3 4 \times + 2 \times"$ would evaluate to 30.

You can think of a statement in this syntax as specifying a binary tree, where each node is an expression. Leaves are numbers and internal nodes are labeled by an operation which is applied to the values represented by the node’s children. For the expression above, we would get the tree:

```
          x
         /
        +
       /
      2
     /
    +
   /
  1 2
```

(a) Give an algorithm which takes as input a statement of length $n$ (as a string) and evaluates it in time $O(n)$. Your algorithm should use work space $O(\text{depth})$, where the *depth* of a statement is the maximum depth of a node in the tree corresponding to the
input (work space is the amount of memory used beyond what is necessary to represent the input).

Hint: what basic data structure makes the most sense for this problem? You do not have to represent the whole tree explicitly.

(b) Give an algorithm which translates a postfix expression into standard parenthesized infix notation (for example, \(1 \ 2 \ + \ 3 \ 4 \ \times \ + \ 2 \ \times\) would be translated to \(((1 + 2) + (3 \times 4)) \times 2\)). For this task it may be helpful to represent the tree explicitly. Your algorithm should use time \(O(n)\).

3. Programming problem on graphs. To be specified in a separate handout.