

## Review of lectures 21 to 26, 30 to 32

The exam is closed book, calculators or laptop computers are not allowed. Good examples of questions are quizzes and homework assigned at the end of each lecture. Also review the answers to the midterm exams.

This sheet contains some preliminary examples of questions about the second part of the course, on the material tested by the second midterm exam, chapters 7, 8, and 10 in the textbook.

1. Chebyshev polynomials  $C_n$ , with  $\deg(C_n) = n$ , are defined recursively as:

$$C_0(x) = 1, \quad C_1(x) = x, \quad \text{for } n > 1: \quad C_n(x) = 2xC_{n-1}(x) - C_{n-2}(x).$$

- (a) Define a recursive function with prototype
 

```
double C ( int n, double x );
```

 that returns the value of the  $n$ -th Chebyshev polynomial at  $x$ .
  - (b) Simulate the call `C(5,1.0)`. Leave the formulas, do not evaluate them.
  - (c) Make a more efficient version of your function via memoization, by storing the results of previous function calls in a vector, maintained by the function. The vector is initialized to hold up to  $n = 50$  when calling the function with  $n = -1$ .
  - (d) Use this stack to give an iterative version of this function.
  - (e) Without using a stack, give an iterative version of this function.
2. To enumerate all numbers divisible by three, we propose a function with prototype

```
void numbers3
( int n, int k, vector<int> accu, int radix, int sum );
/* Writes all possible numbers of n digits long, every digit is in
   the range 0..radix-1, where the sum of the digits is divisible 3.
   The parameter k controls the recursion depth. */
```

Give a recursive definition of this function.

3. To enumerate all possible license plates, we propose a function with prototype

```
void plates ( int n, int k, string accu );
/* Generates and prints all license plates of n characters,
   accumulating in accu. The k controls the recursion depth. */
```

Give a recursive definition of this function.

4. Use a tree to convert the expression  $3*a/(2 + b/4)/c + (5*a)*(b-2)$  into postfix form.
5. Create a Huffman code for the message "structure".
  - (a) Give code to use the appropriate data structure of the STL to make a frequency table for the characters in the message.
  - (b) Given the frequency table, make the Huffman code.  
Draw all intermediate stages in the creation of the Huffman code.
  - (c) Show how to encode and decode the message.

6. Consider the sequence 18 12 3 21 33 27 30.
  - (a) Insert the elements in the sequence into a binary search tree.
  - (b) Delete 21 from the binary search tree.
  - (c) Delete 18 from the binary search tree.
7. What is open addressing? What is chaining?  
Explain the differences between open addressing and chaining.  
Give an example of both techniques to illustrate the differences.
8. Consider the sequence 43 19 12 31 62 81 77 10.
  - (a) Sort the numbers in ascending order with bubble sort. Show all intermediate passes.
  - (b) Sort the sequence with quicksort. Show all intermediate stages in the sorting algorithm.
  - (c) Show how to sort the numbers, while selecting each time the smallest element in the sequence and swapping if necessary.
  - (d) Sort the sequence by inserting in a binary search tree.
  - (e) Show how to build the heap in heapsort. Once the heap is built show how removal produces a sequence in ascending order.
  - (f) Apply Shell sort to the sequence, using values 5,3, and 1 for the gap.
  - (g) Show how to apply merge sort to this sequence.
  - (h) Sort the sequence with insertion sort.
9. Give the code to compute the balance of a tree, using the definitions

```
typedef struct node tree;
struct node
{
    int info;
    tree *left;
    tree *right;
};
```

and the prototype

```
void write_balance ( int k, tree *t );
/* writes the info and balance of every node in inorder traversal,
   k controls the depth of the recursion (call with k == 0) */
```

**The final exam happens on Monday 11 December, from 8AM till 10AM, in 337 BSB.**

In case of a scheduling conflict with another final exam, please let me know as soon as possible so we can schedule a make up.