

Review of the lectures 21 to 35

The exam is closed book, calculators and laptop computers are not allowed. Good examples of questions are quizzes and homework assigned at the end of each lecture.

This sheet contains some preliminary examples of questions which may help you prepare for the second midterm exam.

The material in lectures 21 to 35 covers chapter 7 (recursion), 8 (trees), 9 (sets and maps), 10 (sorting), and 11 (balancing trees). The list of questions below is ***NOT*** exhaustive. The goal of the list is to give you an impression of how much (or how little) you know of the course materials.

1. Give a recursive definition of a function `CharCount` which takes on input a string `s` and a character `c`. The function returns the number of times the character `c` occurs in the string `s`. You must use recursion.
2. The value of the n -th Chebyshev polynomial $T(n, x)$ is recursively computed as follows

$$T(0, x) = 1, \quad T(1, x) = x, \quad T(n, x) = 2xT(n-1, x) - T(n-2, x), \text{ for } n \geq 2.$$

- (a) Define a recursive function `T` which takes on input an integer value for n and a double for x and then returns the value of $T(n, x)$.
 - (b) Explain why the straightforward recursive definition is wasteful. Write an efficient recursive function using memoization.
 - (c) Use a stack to write an iterative version of the function `T`, recursively defined above.
3. Consider a vector `W` of strings, which contains 3 weather type predictions, "`summy`", "`cloudy`", and "`rain`". Each weather type may occur with probability 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, or 0.1. The probabilities are stored in `P`, a vector of doubles.

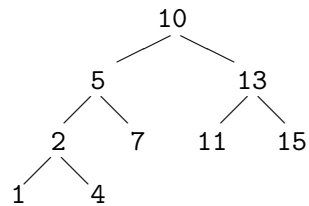
The string "`90% sunny, 10% cloudy`" corresponds to the prediction of 90% sunny and 10% cloudy weather. The sum of the probabilities in the prediction always equals 1.0, or 100% as shown in the prediction.

Design a C++ function `weather` which takes on input the vectors `W` and `P`, to write all possible weather predictions, starting as below:

```
90% sunny, 10% cloudy
90% sunny, 10% rain
90% cloudy, 10% rain
80% sunny, 20% cloudy
80% sunny, 20% rain
80% sunny, 10% cloudy, 10% rain
80% cloudy, 20% rain
etc ...
```

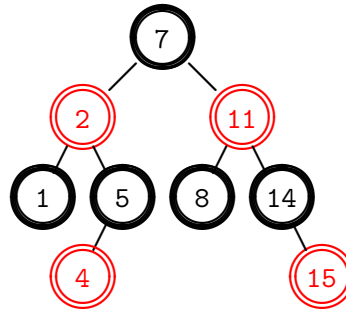
- (a) Write the prototype of the function `weather` and document all input parameters of the function.
- (b) Give the definition of the function `weather`.

4. Use a tree to convert the expression $(5*a + b)/(2*c - b) + a*b$ into prefix form.
5. Consider the binary search tree:



- (a) Give a sequence of the numbers in the tree in which the numbers were inserted to give the tree of the given shape. Show all intermediate trees in the insertion of the sequence.
- (b) Write pseudo code to remove one item from the tree. Illustrate with an example, e.g.: remove 5 or 13 from the tree drawn above.
6. Create a Huffman code for the message "send more money".
- (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.
7. Consider the sequence 49 59 15 41 22 59 78 20.
- (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.
8. Insert the numbers 64 30 25 33 70 83 30 46 90 52 64 50 28 14 54 77 into an AVL tree. Draw all intermediate trees.

9. Consider the red-black tree



- (a) Draw the equivalent 2-3-4 tree.
 (b) Show how to insert 16. Draw all intermediate stages, double circling red nodes.
10. Insert the numbers 64 30 25 33 70 83 30 46 90 52 64 50 28 14 54 77 into a B-tree of 5-nodes. Draw all intermediate trees.

Please note the policy on skipping exam holds: If an exam is missed, then greater weight will be placed on the final exam, especially on the material covered on the missing exam. **What this means is** that if you decide not to take one midterm exam, your final exam will be weighted for one hundred points more.

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