Fundamental Computer Science I
Advanced Placement Exam
October 1st, 2003

Course Fundamental Computer Science, Dr. Holger Kenn
e-mail: h.kenn@iu-bremen.de, tel.:+49 421 200 3112

Name:

Matriculation Number:

INSTRUCTIONS: Write your name on **both** the exam sheet and on the blue book. Both have to be handed in together with all paper used. Read **all** the problems carefully before you start working. Start with the simple problems. Most problems can be answered in a few lines of text or equations. Don't get stuck with the algorithm writing tasks, first try to get an idea how the algorithm works and sketch it for yourself in plain text. Leave the detailed writing of the longer algorithms until the end.

1.) Algorithms

Given this algorithm in pseudocode that multiplies two $n \times n$ matrices A and B.

```
MATRIXMULTIPLY($A, B$)
$n \leftarrow rows[L]$
let $C$ be an $n \times n$ matrix.
for $i \leftarrow 1$ to $n$ do
    for $j \leftarrow 1$ to $n$ do
        $c_{ij} \leftarrow 0$
        for $k \leftarrow 1$ to $n$ do
            $c_{ij} \leftarrow c_{ij} + a_{ik} \cdot b_{kj}$
        end for
    end for
end for
return $C$
```

a) What is the running time of this algorithm in asymptotic notation? Give all bounds.

b) Modify the algorithm so that it multiplies a $m \times n$ with a $n \times p$ matrix. What is the runtime (in asymptotic notation) ?

Given a chain of matrices $< A_,1A_2, \ldots, A_n >$, we can compute the product $A_1A_2 \dots A_n$ either as $(A_1(A_2(A_3 \dots A_n)))$ or as $(A_1A_2)(A_3 \dots A_n)$ or as $(A_1(A_2A_3))(A_4 \dots A_n)$ and so on, each of these matrix multiplications using a different number of scalar multiplications if the matrices $A_1 \dots A_n$ have different sizes, i.e. numbers of rows and colums.

c) Give an algorithm that computes the optimal cost (i.e. the number of scalar multiplications) for a given chain of matrices. Hint: Use dynamic programming.

d) Give an algorithm to extract the optimal parens-configuration from the output and the internal datastructures of your algorithm given in c).

2. Sorting

a) How can Quicksort be modified in order to have a worst-case runtime in $O(n \lg n)$?

b) Why is it necessary to use a stable sorting algorithm as as subroutine of RADIXSORT?

3. Heaps

Given the array $A = (21, 49, 35, 7, 63, 28, 70)$

a) Is this a min heap? Give all violations of the heap property.

b) Give the heap after BUILDMINHEAP has been called and all intermediate steps of BUILDMINHEAP.

4. Hashing

Given an Array of integers of length 16 i.e. with array indices 0 to 15 and a hash function $h(k, i) = ((k \mod 23) + i) \mod 16$.

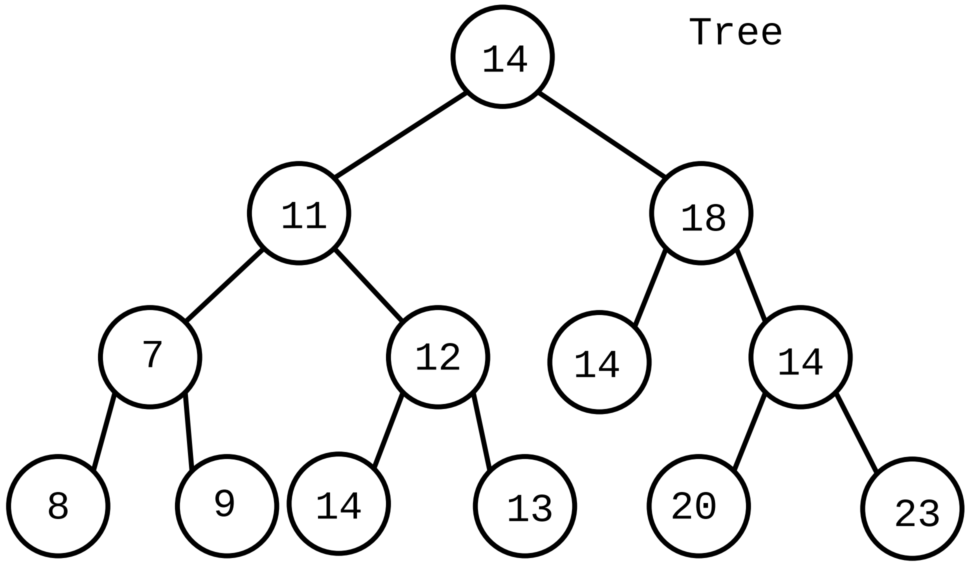
Insert the following sequence of numbers into the array, show the contents of the array after each step. Use hashing with open adressing.

272 273 325 498 15 137 1033

5. Binary Trees

For reference, you will find TREEINSERT below.

Given the following binary tree:



a) Is it a binary search tree? List all violations of the binary search tree property.

b) Give a recursive algorithm for insertion into a binary search tree.

Given an *new, empty* binary search tree.

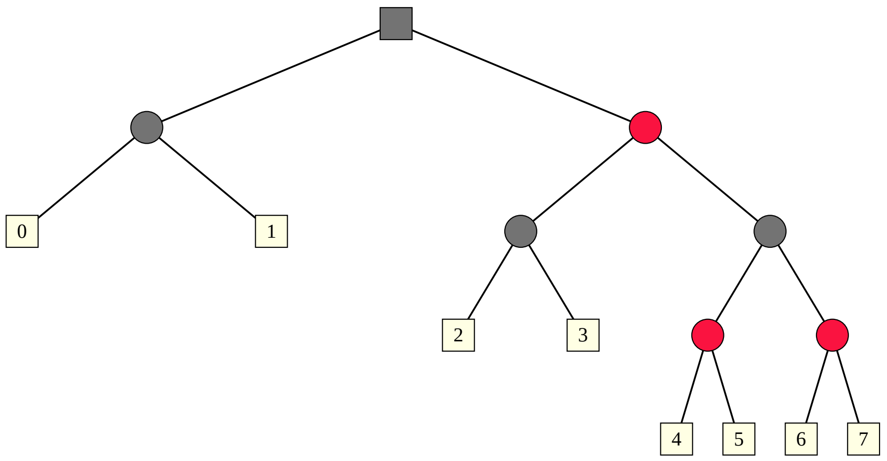
c) Insert the values 12, 15, 3, 7, 15, 12 into the binary search tree. Draw the tree after each insertion.

```
TREEINSERT$(T, z)$
 1: $y \leftarrow$ NIL
 2: $x \leftarrow root[T]$
 3: **while** $x \neq$ NIL **do**
 4:     $y \leftarrow x$
 5:     **if** $key[z] < key[x]$ **then**
 6:         $x \leftarrow left[x]$
 7:     **else**
 8:         $x \leftarrow right[x]$
 9:     **end if**
10: **end while**
11: $p[z] \leftarrow y$
12: **if** $y =$ NIL **then**
13:     $root[T] \leftarrow z$
14: **else**
```

```
        if $key[z] < key[y]$ then
            $left[y] \leftarrow z$
        else
            $right[y] \leftarrow z$
        end if
    end if
```

6. Red-Black-Trees

Given the following red-black tree created by the successive insertion of $0 \ldots 7$ into an empty red-black tree:



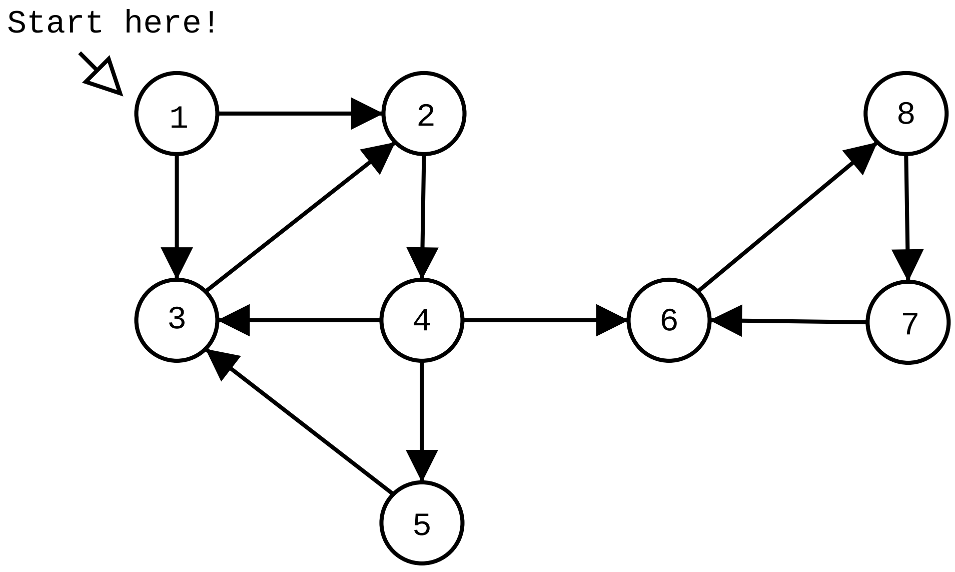
(This type of tree is storing the data elements only in the leaves. The inner nodes carry keys too, but these are only used for internal operations of the datastructure, e.g. searching or rebalancing the tree.)

Draw the tree after the next step, i.e. after the insertion of 8. Give the sequence of rotation operations that are executed and on what nodes and why they are executed, i.e. what red-black property violation is found by the algorithm.

➤

7. Graphs

Given this directed graph.

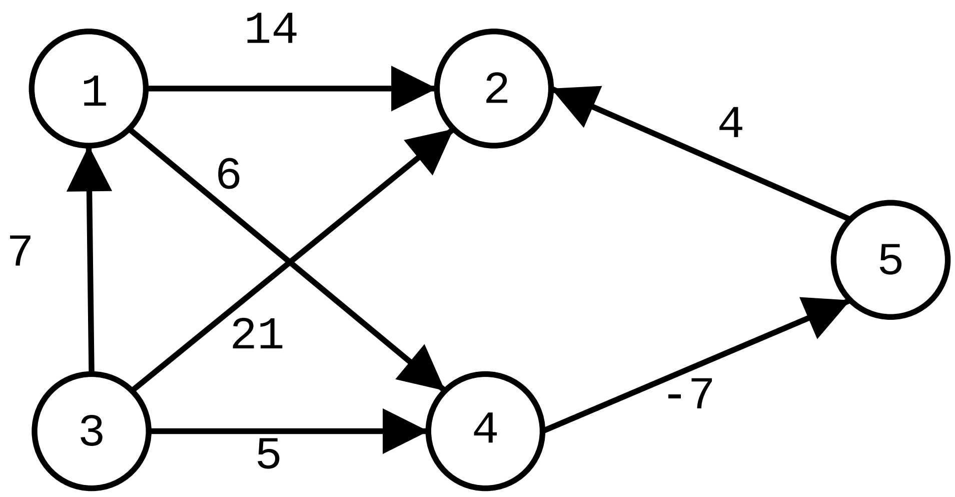


a) What algorithm would you use (and modify) to count and list all back edges in the graph? What would be the modification?

b) Demonstrate this algorithm on the given graph. Start with node 1 and analyzes outgoing edges in clockwise order starting from the incoming edge used by the algorithm. Indicate for all edges at which step of the algorithm an edge is classified and give its classification.

8. Paths

Given the following graph:



a) Give the adjacency (weight) matrix representation of the graph.

b) Determine all-pairs shortes path by demonstrating the Floyd-Warshall-Algorithm on this graph. Give all the intermediate $D^{(i)}$ matrices. Draw a complete graph with the shortest distances as edge weights.

9. Dijkstra's Algorithm

a) What does Dijkstra's Algorithm do? What are the requirements to be able to apply it?

b) What is its runtime? (Hint: consider different implementations of the datastructures and different characteristics of the input.)

10. Few Stops

Given the position of gas stations along a particular route in kilometers from the start point, a trip length t and a maximum distance d that your car can travel with one tank full of gas, give an algorithm that determines the gas stations to stop at while traveling that route in order to do as few tank stops as possible. Prove that your algorithm yields the optimal solution.

11. Binary Counting

Give a proof that incrementing a binary counter costs $O(1)$ worst-case by using one type of amortized analysis.