

# Data Structures and Algorithms

## Exercise 7

Markus Innerebner, Romans Kasperovics  
dsa(a)inf.unibz.it

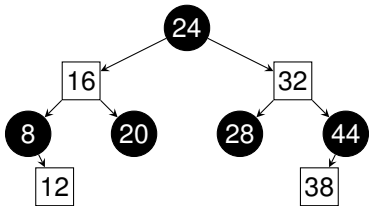
Free University of Bozen - Bolzano, Italy

Thursday, April 9th, 2009

# Assignment 07

## Task 1

Consider the Red-Black tree given below. Circled nodes 24, 8, 20, 28, 44 are black. Rectangular nodes 16, 32, 12, 38 are red. Perform the following operations on the given tree. For each operation start from the original tree, draw all intermediate and resulting trees and specify which of the cases you applied.



- Insert 40
- Delete 20
- Insert 13
- Delete 32

Implement in C the following operations for the hash table abstract data type that uses double hashing to resolve collisions:

- **int\*** `init(int m)` returns a new hash table allocated for  $m$  empty slots;
- **int** `hashval(int key, int i)` returns the slot number for *key* ( $i$ -th probe), i.e.,  $h(k, i) = (h_1(k) + i \cdot h_2(k)) \bmod m$ , where  $h_1(k) = k \bmod m$  and  $h_2(k) = (2k + 1) \bmod m$ ;
- **int** `insert(int H[], int m, int key, bool* inserted)` inserts *key* into the hash table and returns the slot number at which it was inserted;
- **int** `search(int H[], int m, int key, bool* found)` searches *key* in the hash table and returns its slot number;
- **int** `delete(int H[], int m, int key, bool* deleted)` deletes *key* in the hash table (i.e., marks the slot as deleted) and returns its slot number.