08.71.23/24 Tölvunarfræði 2/2a

Final exam **April 30. 2002**

Teacher: Hjálmtýr Hafsteinsson **Time: 9^{00} ? 12^{00}**

The first 5 problems are for all students (both from Tölvunarfræði 2 and 2a) Problem 6 is only for students in Tölvunarfræði 2, but problem 7 is only for students in Tölvunarfræði 2a (engineering students). In both cases ***five best problems out of six count****. All problems have the same value.*

All written materials and a calculator are allowed.

?? Note that when asked to "Describe" or "Show" then it is enough to do that in words and with drawings. If you are to write C++ code you will be asked for that specifically.

?? Give supporting arguments for all answers and remember that it is not necessary to write up definitions from the book.

1. In the weighted versions of Union-Find two trees are merged based on the number of nodes in them, not their height. It is though the height of the trees that determines the worst case time of the Find() operation.

Why isn't the height used in fast Union-Find algorithms? Explain in detail what the problem is, e.g. with examples or drawings.

2. You are given a doubly linked list, where the node pointer `haus` points to the first node. Also given are two node pointers, `p` and `x`. You are to write a piece of code in C++ that moves the node that `p` points to, behind the node that `x` points to, but doesn't change anything else about the list.

a) Below is a pice of code that is claimed to handle the general case. Is that right? Give an argument for that. Fix the code if it is wrong.

```
p->prev->next = p->prev;
p->next->prev = p->next;
p->next = x->next;
x->next = p;
```

b) Show all the special cases that can come u, that the general case does not handle correctly. Show a piece of C++ code to handle each case.

3. Write a function in C++ that gets as parameter a pointer to the root of a binary tree. The function is to return 1 (true) if the tree is a **binary search tree**, but 0 (false) otherwise.

4. Assume that we want to look for a particular value in a *heap.*

a) Implement in C++ a function that starts at the top of the heap and looks only as far down the tree as necessary to either find the value or make sure it is not in the tree. The worst case time for this method will be *O(N)*.

b) Can we use binary search inside the tree to improve the speed of the search? Explain the method or the difficulty with doing it.

5. In *open addressing* hashing the items are placed into the hash table. When there is a collision in table location i we need to find another location for the new item. A special case of open addressing is *linear probing*. There we look first at location i+1, then location i+2, i+3, etc. This method results in clustering.

Implement in C++ another method, where if there is a collision we next check locations i+1, i+4, i+9, ..., in general $(i+k^2)$ mod M, where M is the table size. Show insertion (`insert`) and search methods (`search`) in C++. Also explain the advantages and disadvantages of this method compared to *double hashing.*

Only for students in Tölvunarfræði 2:

6. *Binary insertion sort* is just like insertion sort, except that it uses binary search to find the correct location of the next item in the sorted part of the array.

a) Describe this sorting method in more detail, e.g. with pseudo code (not necessarily C++).

b) When would this sorting method be faster than ordinary insertion sort? Explain.

Only for students in Tölvunarfræði 2a (engineering students):

7. *a)* Is there a 4-node binary tree that has the same *inorder* and *preorder*? Argue that there isn't or show an example of such a tree.

b) What about a 4-node binary tree with the same *preorder* and *postorder*? Give an argument or an example.

c) What about a 4-node binary tree with the same *postorder* and *levelorder*? Give an argument or an example.