08.71.23/24 Tölvunarfræði 2/2a

Final exam **May 3rd, 2004**
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 ***the five best problems out of six count****. All problems have the same value.*

All written material and a calculator are allowed*.*

- Please note that when asked to "Describe" or "Show", then it is enough to do that using words and 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. Write a function in C++ that receives two pointers to singly linked lists, (`node *s` and `node *t`). Your function should return a pointer to a list that consists of nodes alternately from then list pointed to by `s` and the one pointed to by `t` (sort of like a zipper). The first node should come from the `s`-list, and if one list is longer than the other then its last nodes will just be added to the end of the output list. You are not to create new nodes, only change the `next`-pointers such that the lists are combined.

2. Assume a singly linked list, where each node contains an integer. You are to write a recursive function in C++ to calculate the sum of squares, i.e. $(a_1)^2 + (a_2)^2 + ... + (a_n)^2$, where a_i is the integer in node i in the linked list.

a) Consider the functions below and explain why they do not solve this problem. Describe what is wrong in each instance and say what would happen if the functions were to be executed.

i)
```
int sumKvrt( node *h )
{
    if( h != NULL )
        return ( sumKvrt( h->next ) * sumKvrt( h->next ) );
}
```

ii)
```
int sumKvrt( node *h )
{
    if( h == NULL )
        return 0;
    return ( h->item * sumKvrt( h->next ) );
}
```

iii)
```
int sumKvrt( node *h )
{
    return 0;
    if( h->next != NULL )
        return ( h->item * h->item ) + sumKvrt( h->next );
}
```

b) Write a correct recursive version of this function and explain how your function calculates the sum of squares of the linked list.

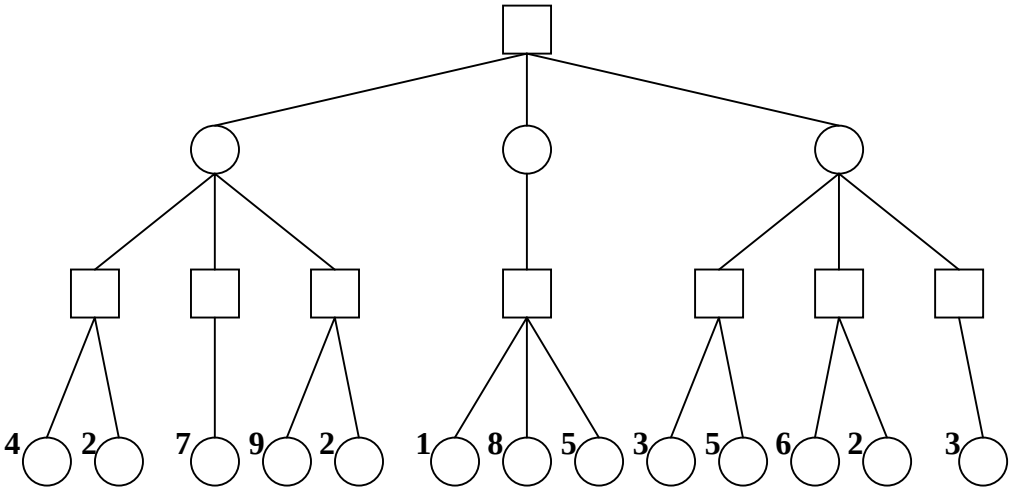
3. In the array implementation of a queue, on page 171 in the textbook, the size of the allocated array is 1 greater than the maximum number of elements that can be put into the queue.

a) Explain the purpose of this one extra array position.

b) Describe a way to implement a queue where the size of the array can be the same as the maximum number of elements.

c) Implement such a queue in C++, i.e. one with the number of array positions the same as the maximum number of elements that will go into the queue.

4. Below is a game tree where the evaluation function has been applied to all positions on depth 3. Remember that a box-node means maximization and circle-node means minimization.



a) Finish calculating the tree according to the Minimax method. Draw the tree and put values on all the nodes (i.e. game positions) that do not have a value in the above tree.

b) If we used the Alpha-Beta version of Minimax then some branches would not have to be calculated. Show where such Alpha-Beta cuts appear in this tree. Compare the number of calculated nodes in the Alpha-Beta version to the Minimax version in part a).

c) Does the order of the children make any difference for Alpha-Beta? If so, show an example of that in the above tree and how much it matters. If it does not matter, explain why.

5. Remember that an *N* item heap stores those *N* items in positions 1 thru *N* of an array.

a) Explain carefully (in words and/or drawings) a method to delete the item in position *i* of the heap. Calculate the time that your method takes.

b) Explain carefully a method to insert a new item into position *i* of the heap. Calculate the time that your method takes.

Only for the students of Tölvunarfræði 2:

6. *a)* Explain why Quicksort is not a stable sorting method.

b) Show how you could change Quicksort so that it becomes stable. Can it be done without any effect on the time complexity of Quicksort? Justify your answer carefully.

Only for the students of Tölvunarfræði 2a (engineering students):

7. Below you are given the results of an inorder, a preorder, and a postorder traversal of a 9 node binary tree, where the nodes have the values **A** thru **I**. Unfortunately some of the values have been lost in transmission (denoted by **?**). Try to construct the binary tree from this incomplete information and describe your reasoning at each step. If the information is not sufficient to construct a unique binary tree, describe the trees that are possible.

Inorder: **IC?HBE?DA**

Preoder: **?C???H?D?**

Postorder: **IHGB??DAE**