

Faculty of Science
FINAL EXAMINATION

COMPUTER SCIENCE COMP 250
INTRODUCTION TO COMPUTER SCIENCE

Examiner: Prof. Michael Langer
Associate Examiner: Mr. Joseph Vybihal

April 28, 2011
2 P.M. – 5 P.M.

STUDENT NAME: _____ **ID:** _____

Instructions:

The exam is 14 pages, including this cover page.

There are 11 questions, worth a total of 30 points.

Answer all questions on the exam question sheet.

The 14 pages includes two extra pages at the end. If you need more space for one of your answers, indicate this by writing, for example, “see page 13 for more”.

No calculators, notes, or books are allowed.

Question	Points	Grade
1	3	
2	2	
3	3	
4	2	
5	3	
6	2	
7	3	
8	2	
9	3	
10	3	
11	4	
Total	30	

1. (3 points)

- (a) Perform the multiplication
- 101.1×0.11
- on the two given
- binary*
- numbers.

Your answer must be in binary. You are not allowed to convert the numbers to decimal.

- (b) Suppose
- i
- is a 16 bit positive integer, represented in binary as bits
- $b_{15} b_{14} \dots b_2 b_1 b_0$
- .
-
- What is the value of the function

$$h(i) = (i/8) \bmod 32 ,$$

written in terms of these bits.

Solution

- (a) 100.001 (1 point)

- (b)
- $b_7 b_6 b_5 b_4 b_3$
- (2 points)

We gave 1 point for $i/8$ (chop off $b_2 b_1 b_0$) and 1 point for the mod (chop off $b_{15} \dots b_8$).

(Several students incorrectly treated $i/8$ as if i were a float rather than an int and gave an answer: $b_7 b_6 b_5 b_4 b_3 . b_2 b_1 b_0$.)

2. (2 points)

Consider the Java code:

```
for (int i=0; i < list.size(); i++){
    System.out.println( list.get(i).toString() );
}
```

- (a) Give an expression for the computation time, in terms of the number of elements in the list, i.e. `list.size()`, if the list were of type `LinkedList<E>`. Justify your answer.
- (b) The following Java code uses the `LinkedList<E>` iterator.

```
for (E e : list){
    System.out.println( e.toString() );
}
```

Would your answer to (a) change if you used this “enhanced for loop” instead?
Why or why not?

Solution

- (a) The standard implementation of `get(i)` is to traverse the list from the beginning to find element `i`. Thus, the time to get element `i` is proportional to `i`. Thus, the loop takes

$$1 + 2 + 3 + \dots + size = \frac{size(size + 1)}{2}$$

operations.

- (b) Yes, it would. The enhanced for loop uses an iterator to traverse the list. So the loop would take time proportional to `size`.

3. (3 points)

- (a) Give the formal definition of “ $t(n)$ is $\Omega(g(n))$ ”.
- (b) Show that $t(n)$ is $\Omega(g(n))$, where

$$\begin{aligned} t(n) &= \frac{1}{5} \log(n-8) \\ g(n) &= \log(n). \end{aligned}$$

Solution

- (a) There exist two numbers c and n_0 such that $c > 0, n_0 > 0$ and, for all $n > n_0$, $t(n) \geq cg(n)$.
- (b) We are looking for a lower bound so let's try some constant $c < \frac{1}{5}$. Let's try $c = \frac{1}{10}$.

$$\begin{aligned} \frac{1}{5} \log(n-8) &> \frac{1}{10} \log n \\ \iff \log(n-8) &> \frac{1}{2} \log n \\ \iff \log(n-8) &> \log \sqrt{n} \\ \iff n-8 &> \sqrt{n} \end{aligned}$$

But the last inequality is true if n is sufficiently large, since n grows faster than \sqrt{n} . We still need to choose an n_0 . The inequality holds for $n_0 = 16$ since $8 > 4$. Moreover, dividing both sides by \sqrt{n} gives

$$\sqrt{n} > 1 + \frac{8}{\sqrt{n}}$$

which holds for all $n > 16$ since the left side is increasing and the right side is decreasing. So, $n_0 = 16$ does the job (and $c = \frac{1}{10}$).

Marking scheme: 1 point for (a). 2 points for (b), with 1 for choosing a good c and 1 for choosing a good n_0 where “good” means properly justified.

4. (2 points)

Solve the following recurrence.

$$t(n) = t\left(\frac{n}{2}\right) + \frac{n}{2} - 2$$

You may assume that $t(1) = 1$ and that n is a power of 2.

Solution

$$\begin{aligned} t(n) &= t\left(\frac{n}{2}\right) + \frac{n}{2} - 2 \\ &= \left(t\left(\frac{n}{4}\right) + \frac{n}{4} - 2\right) + \frac{n}{2} - 2 \\ &= \left(t\left(\frac{n}{8}\right) + \frac{n}{8} - 2\right) + \frac{n}{4} - 2 + \frac{n}{2} - 2 \\ &= \left(t\left(\frac{n}{n}\right) + \frac{n}{n} - 2\right) + \cdots + \frac{n}{8} - 2 + \frac{n}{4} - 2 + \frac{n}{2} - 2 \\ &= t(1) + 1 + 2 + 4 + 8 + \cdots + \frac{n}{2} - 2 \log n \\ &= n - 2 \log n \end{aligned}$$

Marking scheme:

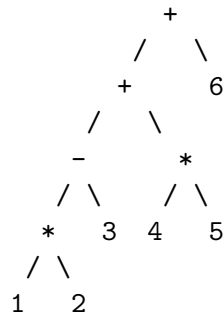
We gave 0.5 points for the $2 \log n$ part. For the “n” part, we gave 0.5 for a series expansion, another 0.5 if you got basically the right answer but you don’t know that $2^{\log n} = n$, and the final 0.5 if you were able to reduce the answer to n .

5. (3 points)

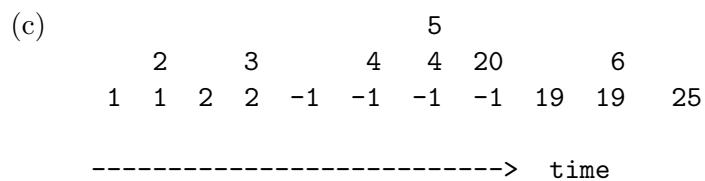
- (a) Consider the infix expression $1 * 2 - 3 + 4 * 5 + 6$.
Assuming the usual operator precedence order, draw this expression as an “expression tree.”
- (b) Write the expression in (a) as a post-fix expression. Your expression must not contain any brackets.
- (c) Show how the stack evolves over time, as the postfix expression is evaluated.
Do not put operators on the stack.

Solution

(a)



(b) $1\ 2\ *\ 3\ -\ 4\ 5\ *\ +\ 6\ +$



6. (2 points)

A tree can be represented using lists, as follows:

```

root      = element
tree      = element | ( root listOfTrees )
listOfTrees = tree   | tree listOfTrees

```

Stated another way, the root is an element; a tree is either an element or else it is a root element followed by a list of its children, each of which are trees. In the latter case, the list is enclosed by open brackets. Finally, a list of trees is either a single tree, or it is a tree followed by a list of trees.

Draw the tree that corresponds to the following:

(1 (2 6 7 8) 3 (4 (9 12)) (5 10 11))

You may use either a “first child, next sibling” representation, or you may use a separate edge from a parent to each of its children.

Solution

Example taken from <http://gajon.org/trees-linked-lists-common-lisp/>

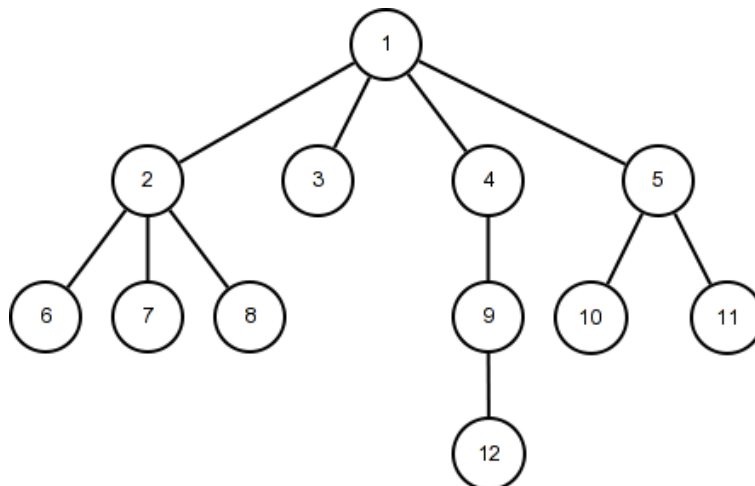


Figure 1: A simple tree.

7. (3 points)

A d -heap is defined as follows. It is a tree that has up to d children per node, and each level of the tree is full except possibly for the last level. In the last level, nodes are filled from the left. The heaps we saw in the lectures were 2-heaps.

- (a) Give a precise parent-child indexing scheme for representing the elements of a 4-heap using an array. Unlike in the 2-heap case we saw in class where the root was at array element 1, here I am requiring that the root is at array element 0.

Note: Questions (b) and (c) do *not* require that you answer (a) correctly.

- (b) Consider a 4-heap with ten elements, $a[i] = i$, for $i = 0, 1, \dots, 9$. What is the result of applying `removeMin()` to this 4-heap? Note that you need to generalize the `removeMin()` method from class which applied only to 2-heaps.

You may write your answer either as a tree or as an array.

- (c) Give an upper bound on the maximum number of nodes in a d -heap of height h . Do *not* leave your answer as a summation.

Solution

- (a) `child = 4 *parent + i`, where $i = 1, 2, 3, 4$
`parent = (child - 1) / 4`

- (b) 152349678

- (c)

$$1 + 4 + 4^2 + 4^3 + \dots + 4^h = \frac{4^{h+1} - 1}{4 - 1}$$

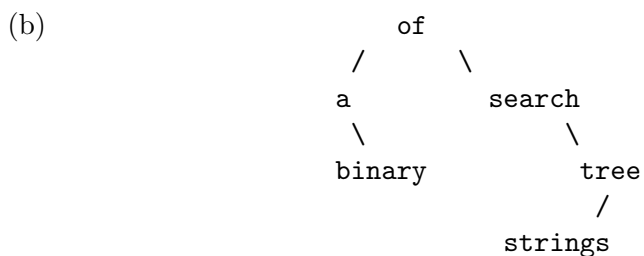
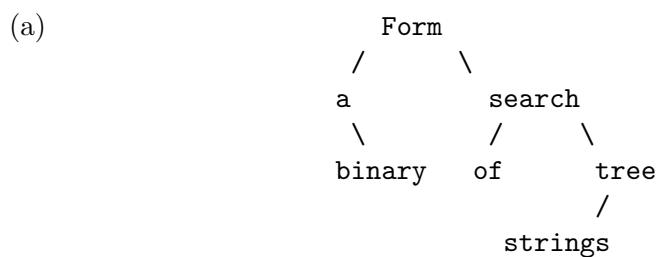
8. (2 points)

- (a) Form a binary search tree of strings from the sentence:

Form a binary search tree of strings

In particular, the tree must be constructed by adding the words in the order they are found in the sentence, namely first add the word “Form” to an empty tree, then add the word “a”, etc.

- (b) What is the tree that results, if you run
- `remove(‘Form’)`
- on the tree in (a) ?

Solution

9. (3 points)

Suppose you have a map that you wish to represent using either a hash table (HT) or a binary search tree (BST). You may assume that your keys are comparable, and that the BST is roughly balanced namely the leaves are all roughly at the same height.

Compare the runtimes of HT and BST for each of the following operations, and briefly justify your answers.

Do not compare constant factors in the operations e.g. how long it takes to compute a hash value versus comparing two keys. Also do not consider the work that needs to be done for BST rebalancing.

- (a) Remove the entry (k, v) corresponding to a given key k .
- (b) Return all keys that have some given value v .
- (c) Return all entries (k, v) such that $k_1 < k < k_2$, where k_1 and k_2 are given keys.

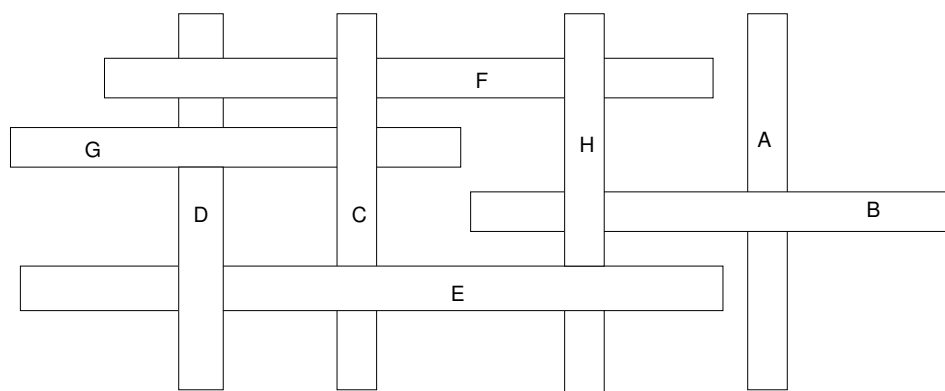
Solution

This question was not handled well, and so we decided not to grade question (c) and give all the points for (a,b) – 1.5 for each.

- (a) HT takes $O(1)$ since a proper hash table will have very few elements at each entry. A balanced BST takes $O(\log n)$, where n is the number of entries in the map since this is roughly the height of the tree.
- (b) Both HT and BST take $O(n)$ since neither data structure is organized by value. You need to iterate through all n entries in the map. This takes about n steps for either data structure.

10. (3 points)

The figure below shows a set of rectangles whose overlap relationships can be represented using a directed graph, as follows. Each rectangle is represented in the graph by one vertex, and there is a directed edge in the graph whenever part of one rectangle lies on top of another rectangle. For the example below, there is an edge from E to C, but not from C to E and there are no edges between C and D since these rectangles do not overlap at all.



- Give the adjacency list for this graph, such that *the vertices are ordered alphabetically*. Note that this ordering implies that there is a unique answer.
- Using (a), give the ordering of vertices visited in a *breadth first* traversal of the graph, starting from vertex C.
- Using (a), give the ordering of vertices visited in a *pre-order (depth first)* traversal, again starting from C.

Solution

- A -
 - B - A
 - C - F, G
 - D - E
 - E - C, H
 - F - D
 - G - D
 - H - B, F
- CFGDEHBA
- CFDEHBAG

11. (4 points)

- (a) Exactly one of the lines (1)–(4) in the `Test` class has a compiler error. Which one and why?
- (b) Assuming you have commented out the error line identified in (a), what is the output when you run the `Test` class? See lines (1)–(4). Briefly explain why, using the terms “override” or “overload” when appropriate. Answer anywhere on the page.

```
public class Test {
    public static void main(String[] args) {
        Vehicle v          = new Vehicle();
            v.driver      = "Lady Gaga";
        Movable m          = new Car();
        System.out.println( v.move("Eminem") );           // (1)
        System.out.println( v.move() );                  // (2)
        System.out.println( m.move("Snoop Dogg") );      // (3)
        System.out.println( m.move() );                  // (4)
    }
}

public interface Movable {    public String  move();    }

public class Vehicle implements Movable{
    public String  driver = "Rihanna";
    Vehicle(){
    Vehicle(String driver){ this.driver = driver; }
    public String move() { return "vehicle driven by " + driver; }
    public String move(String driver) { return "vehicle driven by " + driver; }
}

public class Car extends Vehicle implements Movable{
    public String move(){ return "car driven by " + driver; }
}
```

Solution

- (a) The error is in line (3). `m` is of declared type `Movable` but the `move()` method of the `Movable` interface doesn't have a `String` argument.
 [NOTE: The question should have said “compiler” error, not “syntax” error. I announced during the exam, and I also gave hint that the error is in lines (1)-(4).]
- (b) (1) prints “vehicle driven by Eminem”, since “Eminem” is the parameter passed to the overloaded `Vehicle.move(String)`
 (2) prints “vehicle driven by Lady Gaga”, since `Vehicle.move()` uses the `Vehicle.driver` field, which has value “Lady Gaga”.
 (4) prints “car driven by Rihanna”, since `Car` objects inherit the `Vehicle.driver` field, which is initialized to “Rihanna”. `Car.move()` overrides `Vehicle.move()`.