

CS312 Preliminary Exam 2
Fall 2007

There are 6 problems on this exam. You have one and a half hours for the exam. This is a closed-book examination; you *may not* use outside materials.

Name : _____

Net ID: _____

Problem	Points	Score
1	16	
2	16	
3	16	
4	20	
5	22	
6	10	
Total	100	

1. True/False [16 pts]

(parts a–h; 1 point off for each wrong answer, 0 points for each blank answer)

- a. ____ A declaration `val empty : 'a stack` in the signature of a stack data structure indicates that the implementation is functional (i.e., the stack is immutable).
- b. ____ Representation invariants should be documented in module interfaces.
- c. ____ During each garbage collection phase, a copying collector traverses exactly half of the memory.
- d. ____ In an AVL tree, the difference in length between the shortest and the longest path from the root to the leaves is at most one.
- e. ____ Hash table operations are performed in constant time if the number of buckets is proportional to the size of the table and the hashing function satisfies the uniform hashing assumption.
- f. ____ In functional languages that use function closures to model higher-order functions and static scoping, it may not be safe to pop function frames off the stack as functions return.
- g. ____ The amortized complexity of an insertion operation into a resizable array with n elements is $O(\lg n)$.
- h. ____ Union-find data structures can unify arbitrarily large equivalence classes in $O(1)$ time.

2. References [16 pts] (parts a–b)

(a) [4 pts] Write a short ML program whose execution produces memory cells that would not be reclaimed by a reference counting garbage collector. Briefly explain.

(b) [12 pts] Right-threaded trees are binary trees where each unused right pointer from a node to a Nil child is replaced with a pointer to the node's in-order successor. This makes it easy to traverse the entire tree by following right links.

Consider the following datatype for binary trees with mutable links:

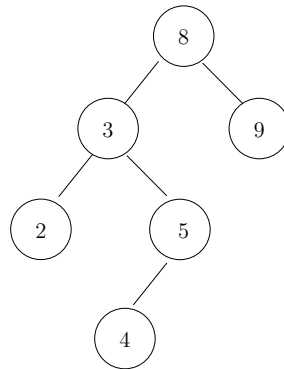
```
datatype 'a tree = Nil | Node of 'a * 'a tree ref * 'a tree ref
```

Finish up the implementation of the function `rthread` below that updates a tree, building all of its right-threaded links. The function updates the existing tree without building any new tree nodes. It returns `SOME x` where `x` is the right link of the rightmost tree node, if any; or `NONE` otherwise.

```
fun rthread(t : 'a tree) : 'a tree ref option =  
  case t  
  of Nil => NONE  
   | Node (_, l, r) => case (rthread !l, rthread !r) of
```

3. **Balanced Trees** [16 pts] (parts a–d)

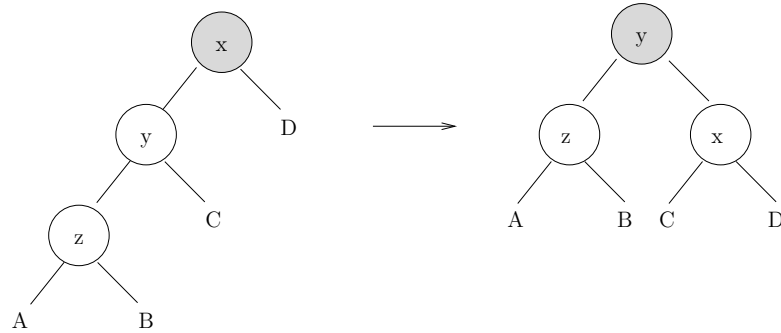
(a) [3 pts] Consider the following binary search tree:



Can this be an AVL tree? Show why or explain why not.

(b) [3 pts] Can the above tree be a red-black tree? Show why or explain why not.

- (c) [6 pts] Consider the operation described in the figure below, where a rotation is used to re-balance a sequence of two red nodes in a red-black tree. In this figure, x , y and z are single nodes, and A , B , C , and D are subtrees. Shaded nodes are black, and clear nodes are red. Nodes x and y are re-colored by this operation.



Is this operation always safe? If yes, argue why. If not, describe the problem, the cases when it happens, and give a solution.

- (d) [4 pts] Explain why the above re-balancing operation would be desirable.

4. Environment Model [20 pts] (parts a–b)

(a) [15 pts] Consider the following program:

```
let val x = ref 1
    fun f(k) = k (!x)
    val x = ref (x := 2; 3)
in
  f (fn z => (print "hello"; z));
  (x, f)
end
```

Draw the environment diagram at the point where the program prints "hello". Clearly indicate the current environment, as well as the environments saved at each function call.

(b) [5 pts] What memory cells that can be collected at the end of this computation?

5. **Types** [22 pts] (parts a–c)

(a) [6 pts] Write a function `f` with the following type:

```
val f: (('a ref) * 'b -> 'c) -> ('a -> 'b -> ('c ref))
```

(b) [4 pts] Write a one-line function for which ML cannot infer a valid type. Briefly explain.

(c) [12 pts] Consider the following SML function for which we want to infer an appropriate type:

```
fun g(x,y,z) = if y 1 then x z else (y, z 2)
```

Write down the set of type constraints for function `g`. For each equation, indicate how it has been derived. Then show the solution types inferred for each of the arguments `x`, `y`, and `z`.

6. Recurrences [10 pts] (parts a–b)

Consider the following recurrence relation:

$$\begin{aligned}T(1) &= 1 \\T(n) &= 3T(n/2)\end{aligned}$$

(a) [5 pts] Show that $T(n)$ is $O(n^2)$.

(b) [5 pts] Find a constant c such that $T(n)$ is $\Theta(n^c)$. Explain.