

CSE 305 Programming Languages Spring, 2010

Homework 11

Maximum Points: 16

Due 10:30 AM, Friday, April 23, 2010

Professor Shapiro

April 16, 2010

Write the answers in a file named `hw10.txt`. Put your name and user name at the top of the file, and submit the file, using the UNIX command, `submit_cse305 hw10.txt`. You will be directed to submit 3 more files below.

Note: When you are asked to write a program, you are never to trivialize the task by using some library program or other predefined program which does the job for you. It is acceptable to research the problem on the Web, or in other resources, but if you do, you must cite the sources that you used. It is never acceptable to claim credit for work that is not your own.

1. (5) Common Lisp has a numerical equality function whose name is `=`, and which takes an arbitrary number of numbers as arguments. For example,

```
cl-user(117): (= 3 (- 5 2) (+ 1 2))
t
cl-user(118): (= 3 (- 5 3) (+ 1 2))
nil
```

Perhaps you would prefer to use `==` as do the C-based languages.

- (a) (3) Write a Common Lisp macro whose name is `==`, and which is a synonym of `=`. Put your definition in a file named `equality.cl`. List that file here, and submit it.
- (b) (2) Put here a transcript of a test of your `==` macro. The following shows how to do this with an interactive Common Lisp shell called from a terminal window:

```
<timberlake:Test:1:288> composer
International Allegro CL Enterprise Edition
8.1 [Linux (x86)] (Aug 21, 2009 10:13)
Copyright (C) 1985-2007, Franz Inc., Oakland, CA, USA. All Rights Reserved.
...
cl-user(1): :ld equality
; Loading ../equality.cl
cl-user(2): <Your test here>
cl-user(3): <Your test here>
cl-user(4): :ex
; Exiting
<timberlake:Test:1:289>
```

Continued on next page.

2. (8) We can represent a binary search tree (bst) in Erlang as follows. (We'll assume that the data in the tree are integers, and no integer is stored more than once.)

- An empty bst is the empty list, `[]`;
- a bst containing only one integer, 10, say, is `[10]`;
- a bst containing more than one integer is the list `[N, Lesser, Greater]`, where `N` is some integer, `Lesser` is a bst containing integers less than `N`, and `Greater` is a bst containing integers greater than `N`.

The exact shape of a bst depends on the order that the integers were inserted. For example, after inserting the integers 8, 10, 5, 10, and 6 into an empty bst, in that order, the bst will be the list `[8, [5, [], [6]], [10]]`.

The nice thing about bsts is that, on average, insertion and search are $O(\log n)$.

You are to write an Erlang module named `bst` containing the functions `insert/2`, and `member/2`. Both functions are to be overloaded functions, dispatching on pattern matching and guards. In each case, you are to use an algorithm that is, on average, $O(\log n)$.

Note:

- If you use a variable in a function header that you don't use in the body, you should use the "don't care" variable, `_`.
- Since Erlang represents strings as lists of integers, a list of integers might print looking like a string. Don't worry about that.

(a) (6) Put your definitions of the `bst` module, containing the functions `insert/2`, and `member/2` in a file named `bst.erl`. Make sure you export the functions `insert/2` and `member/2`. Put the file here and submit it.

(b) (2) Test your `bst.erl` program from the directory where you've stored it, by executing the following two Unix commands:

```
erl -compile /projects/shapiro/CSE305/ProgramsForHomeworks/testBST.erl
erl -noshell -s testBST testBST
```

Place here a transcript of your test run, starting from the compile command.

Continued on next page.

3. (3) Write a Prolog program called `merge` to take three arguments, which should succeed when the first two are lists of the same length, for example `[a,b,c,d]` and `[1,2,3,4]`; and the third list consists of the interleaving of the first two lists, for example `[a,1,b,2,c,3,d,4]`. Put that program in a file called `merge.pro`, print it here and submit it.

Make sure that you can use your `merge` program as follows. However, you needn't include such a transcript in this submission.

```
<timberlake:Test:1:310> prolog
SICStus 4.0.5 (x86_64-linux-glibc2.3): Thu Feb 12 09:48:30 CET 2009
Licensed to SP4cse.buffalo.edu
| ?- ['merge.pro'].
% compiling /projects/shapiro/CSE305/Test/merge.pro...
% compiled /projects/shapiro/CSE305/Test/merge.pro in module user, 0 msec 544 bytes
yes
| ?- merge([a,b,c,d],[1,2,3,4],X).
X = [a,1,b,2,c,3,d,4] ?
yes
| ?- merge([a,b,c,d],X,[a,1,b,2,c,3,d,4]).
X = [1,2,3,4] ?
yes
| ?- merge(X,Y,[a,1,b,2,c,3,d,4]).
X = [a,b,c,d],
Y = [1,2,3,4] ?
yes
| ?- C-d
<timberlake:Test:1:311>
```