CISC280 Spring 2007 Lab 5

The Scheme function **time** allows you to discover the amount of time a procedure takes to execute. You call it by wrapping it around the call to the procedure that you wish to time, e.g.

```
(time (fast-expt 2 100))
```

Code **expt** and **fast-expt** as shown in the book (section 1.2.4).

In this lab you will use **time** to compare the speed (*use CPU time*) of the iterative version of expt and fast-expt. Alas, the time, reported in milliseconds, is short and so you must run many repeats to get noticeable time. Run the procedures enough times to get run times between 0.3 and one second. The following procedure will help: it takes a procedure name and one argument and repeatedly calls the procedure on the argument¹. Type it in and test it to be sure you entered it correctly. Look carefully at the definition and see if you can figure out how to call it without asking your TA, but don't spend too much time on it. What would be a good procedure to test it on (i.e. how can you tell it is working)?

```
;Runs the named procedure on arg, num-repeats times.
(define repeat-proc
  (lambda (num-repeats proc-name arg)
      (cond ((= 0 num-repeats) ())
            (else
               (proc-name arg) ;fyi: how does this line show that
               ;cond differs from if? Also, how
              ;does this proc differ from assert?
              ;Can you write both?
              (repeat-proc (- num-repeats 1) proc-name arg)
              ))))
```

Now modify this procedure to accept one additional argument so that you can call it on expt and fast-expt and the arguments they need.

Use a base of 10, and raise to the following non-random powers: 62, 92, and 128. Experiment to find a number of repeats, n, that takes around 300-1000 msec of CPU time to perform. Run repeat-proc on each data point for n repeats, and do it several times for each data point to get a range of results (consider making

¹Review "map" if you're having trouble thinking about procedures as parameters



a Scheme file to do these runs for you). (If you are submitting, **show** your results by saving your interaction file (as text) or by scripting.

Answer the following questions in the comments of your code file:

- 1. What is O() time for expt, fast-expt? Be sure you discuss this with your TA if you have questions. Remember that we typically equate time with procedure calls.
- 2. Which of the three exponents has the longest time for fast-expt? Explain why, based on how the procedure works. Be specific.

If you need to submit (see syllabus if you are unsure) submit your code file(s) and a script of several well-chosen test cases via MyCourses (due Thursday midnight) and on paper (to your TA's mailbox Friday by 1 p.m.) to receive full credit.

When you use MyCourses, remember that you can "upload" files multiple times, but you only click "submit" once.