Lab 3

Submit either .scm files (for code) or text files (for written answers to questions) for each of the following. All code files for this lab must contain proper tests that run when the file is loaded.

The quote function prevents its argument from being evaluated. So the following expression evaluates to the symbol a, not the value that the symbol is bound to.

```
> (quote a)
a
```

a

eq? is an equality predicate that allows comparison of things besides numbers, in this case symbols:

```
> (eq? (quote a) (car (list (quote a) 2 3)))
#t
```

The single quote character is syntactic sugar for the application of the quote function. In all cases, the single quote applies to the single expression that follows it:

> (eq? '< (cadr '(0 < 1))) #t

Note that the following three expressions return exactly the same structure:

```
> (list '0 '> '0)
(0 > 0)
> (list 0 '> 0)
(0 > 0)
> '(0 > 0)
(0 > 0)
>
```

Be prepared to ask questions about **quote** in class, and to answer questions about **quote** on exams.

Programs

- 1. AS&S 1.11. Answer: what is big O for each version of this function for time? space?
- 2. AS&S 1.30.
- 3. AS&S 1.41
- 4. AS&S 1.42

- 5. In AS&S section 1.3.4, the "rights" of first-class computational elements are listed. Though C++ does not allow functions to be parameters or included in arrays, it does allow function pointers to be both of those things. Does this give C++ the same expressive power as Scheme? What do AS&S say? Keeping in mind the previous two problems, do you agree? Explain why or why not.
- 6. If 2³¹ function calls can be processed each second, calculate the amount of time required to perform the *tree recursive* fib code for (fib 100). Report your answer in sensible units. Show all of your calculations.¹
- 7. AS&S 2.2 (hint: read pages 79-89 to understand what this question is asking. This is *crucial* stuff for this course.)
- 8. AS&S 2.20 (At last! So many cool things can be done with this.)
- 9. Write **sort** as an accumulation.

Submit your code file and a script (or interactions) showing files being loaded (and tested) via Sakai (due Sunday midnight) and on paper (to your TA at the START of **Monday** lab) to receive full credit.

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

 $^{^1\}mathrm{Food}$ for thought: could most computers handle this computation on their stack? Why is the answer "yes"?