CS3304 Program #2 Due: April 22

The POWER Set

The power set of a set 'X' is the set of all subsets of 'X' including the empty set {} and 'X' itself. For example, the power set of the set {1 2 3} is

```
{ {} {1} {2} {3} {1 2} {1 3} {2 3} {1 2 3} }
```

Sets can be represented in LISP by the lists containing their elements. Thus the set {1 2 3} can be represented by (1 2 3).

Write a LISP function, POWER, that takes the representation of a set as an argument and returns the representation of the power set for the argument. For example,

```
(POWER '(1 2 3))
```

returns

```
( () (1) (2) (3) (1 2) (1 3) (2 3) (1 2 3) )
```

A simple but important example is (POWER '()) which is (()).

You can assume that each set is valid and contains only single digit numbers, e.g. (1 2 3). To get 100% on your program you must (a) compute the complete power set, (b) all sets within the power set must be in increasing order relative to the number of elements, (c) sets with same size must be in sorted order and (d) the elements within each set must be sorted. The example above illustrates these requirements. Part (a) is minimally required. Deviation from (b) - (d) will result in a 10 point deduction for each. Make sure that your program generalizes to base sets of length n.

You will turn into me (a) a copy of the input file, including the invocations of POWER, (b) a copy of the file generated by the Scheme interpreter that illustrates valid function definitions and the results of invoking POWER, and (c) the source code for your program, and (d) a README file

Run your program on the following sets:

```
'()
'(1)
'(1 2)
'(1 2 3)
'(1 2 3 4)
```