Lauren Schmidt
CSC313
Miscellaneous Public Methods Research Page

Table.sort:
A useful function on arrays is table.sort. It receives the array to be sorted, plus an optional order function. This order function receives two arguments and must return true if the first argument should come first in the sorted array. If this function is not provided, sort uses the default less-than operation (corresponding to the `<´ operator).

On each iteration, the **for** loop calls its iterator function with two arguments: the invariant state and the control variable. A stateless iterator generates the next element for the iteration using only these two arguments. A typical example of this kind of iterator is ipairs, which iterates over all elements in an array, as shown:

```
a = {"one", "two", "three"}
 for i, v in ipairs(a) do
   print(i, v)
 end
```

The state of the iteration is the table being traversed (the invariant state, which does not change during the loop), plus the current index (the control variable). Both ipairs and the iterator it returns are quite simple; we could write them in Lua as follows:

```
function iter (a, i)
  i = i + 1
  local v = a[i]
  if v then
    return i, v
  end
end

function ipairs (a)
  return iter, a, 0
end
```

The pairs function, which iterates over all elements in a table, is similar, except that the iterator function is the next function, which is a primitive function in Lua:

```
function pairs (t)
  return next, t, nil
end
```

Resource: https://www.lua.org/pil/7.3.html