CHAPTER 9 (DAY 7) EXERCISE

Selection Sort

This exercise is based on the problem of sorting a list of numbers, which is one of the classic computing problems. Note that R has an excellent sorting function, **`sort(x),`** which we will not be using.

To judge the effectiveness of a sorting algorithm, we count the number of *comparisons* that are required to sort a vector x of length n. That is, we count the number of times we evaluate logical expressions of the form **`x[i] < x[j].`** The fewer comparisons required, the more efficient the algorithm.

The simplest but least efficient sorting algorithm is selection sort. The selection sort algorithm uses two vectors, an unsorted vector and a sorted vector, where all the elements of the sorted vector are less than or equal to the elements of the unsorted vector. The algorithm proceeds as follows:

1. Given a vector **`x`**, let the initial unsorted vector **`u`** be equal to **`X`**, and initial sorted vector **`s`** be a vector of length 0.
2. Find the smallest element of **`u`** then remove it from **`u`** and add it to the end of **`s`**.
3. If **`u`** is not empty then go back to step 2.

Write an implementation of the selection sort algorithm. To do this you may find it convenient to write a function that returns the *index* of the smallest element of a vector.