Computer Engineering 12
Project 6: A Quick Sorting Assignment

Due: Sunday, June 9th at 11:59 pm

1 Introduction

Professor Loony has decided to take it easy on you since final exams are next week. You only have to implement a sorting algorithm that you've already learned in class, and you'll be modifying his implementation of a set ADT.

2 Interface

Only one interface function must be modified for this assignment:

- `void *getElements(SET *sp);`
  allocate and return a sorted array of elements in the set pointed to by *sp*

3 Implementation

As required by Professor Loony, you will modify the `getElements` function to return a sorted array of elements in the set. He requires you to use the ***quicksort*** algorithm, which was discussed in class and in the text. Quicksort is a recursive sorting algorithm that is very fast in practice. The basic algorithm works as follows:

1. Choose a value from the subarray to be sorted. This value is called the pivot. The choice of pivot does not affect the correctness of the algorithm, but can affect its efficiency.
2. Partition the subarray around the pivot so that the pivot is in the correct location, all values to the left of the pivot are less than the pivot, and all values to the right of the pivot are at least the value of the pivot.
3. Recursively sort the subarrays to the left and right of the pivot.

Note that you are required to implement the sorting algorithm yourself and cannot just call the `qsort` library function to do the work for you. Professor Loony may be crazy, but he's not stupid!

4 Submission

Download the `project6.tar` file from the course website to get started. Modify the `table.c` source file. Submit a `tar` file containing the `project6` directory using the online submission system.

5 Grading

Your implementation will be graded in terms of correctness, clarity of implementation, and commenting and style. Your implementation ***must*** compile and run on the workstations in the lab. The algorithmic complexity of each function in your abstract data type ***must*** be documented.