Activity 1. Sorting Algorithm Behavior

In computer science, the algorithm is a very important concept and can be seen as a model of the operations done by a computer to solve some problem, for example, sorting. A very important variable of interest in analyzing a sorting algorithm's behavior is the number of comparisons that need to be made before a list is sorted. This is a random variable. Mathematical expressions for the expected number of comparisons can be obtained.

For those unfamiliar with sorting algorithms, we review the workings of two very popular ones considered in this activity: the quicksort and the bubble sort. More detailed information can be obtained by checking the book: Ross, S.M.(2002). "Probability Models for Computer Science." Harcourt Academic Press. The data in this activity come from a simulation of the workings of the bubble sort algorithm. If you are interested in seeing the R program that I used to create these data, read Sanchez, J (2003) "Data Analysis Activities and Problems for the Computer Science Major in a Post-Calculus Introductory Statistics Course," in the 2003 Proceedings of the American Statistical Association, Statistical Education Section [CD-ROM], Alexandria, VA, ASA.

The problem

One thousand random permutations of the list 2, 70, 11, 47, 75, 100, 84, 32, 42, 43, 34, 22 were sorted by 1000 users using the same computer, and X = the number of pairwise comparisons needed to sort the list were recorded. We don't know what kind of sorting algorithm was used by the computer to do the sorting, but we know that it was either the bubble sort or the quicksort algorithms discussed earlier in class when talking about the Expected Value. Use the records provided in the data set "sort" to determine which of the sorting algorithms was used by the computer. Support your answer with a thorough descriptive and inferential data analysis. Note: The data set with the records is called *sort* and can be found at: `http://www.stat.ucla.edu/ jsanchez/ oid03/index.html` under the heading "Data Sets."

Review of sorting algorithms as applied to this problem

Ross (2000) p. 8, describes the Bubble Sort Algorithm as follows: Suppose we are given a set of r distinct values $y_1, y_2, ..., y_r$ that we desire to put in increasing order or, as is commonly called, to sort them. The *bubble sort* is an algorithm that can be used. Starting with any initial ordering, it sequentially passes through the elements of this ordering, interchanging any pair that it finds out of order. That is, the first and second values are compared, and interchanged if the second is smaller; then the new value in second position is compared with the value in the third position, and these values are interchanged if the latter is smaller than the former; then the new value in the third position is compared with the value in the fourth position, and so on until a comparison is made with the final value in the sequence, and an interchange, if necessary, is made. At this point the first pass through the list is said to have occurred. This process is then repeated for the new ordering, and this continues until the values are sorted. For instance, if the initial ordering of values is

2,70,11,47,75,100,84,32,42,43,34,22

then the successive orderings in the first pass through are as follows:

2,70,11,47,75,100,84,32,42,43,34,22
2,11,70,47,75,100,84,32,42,43,34,22
2,11,47,70,75,100,84,32,42,43,34,22
2,11,47,70,75,100,84,32,42,43,34,22
2,11,47,70,75,100,84,32,42,43,34,22
2,11,47,70,75,84,100,32,42,43,34,22
2,11,47,70,75,84,32,100,42,43,34,22
2,11,47,70,75,84,32,42,100,43,34,22
2,11,47,70,75,84,32,42,43,100,34,22
2,11,47,70,75,84,32,42,43,34,100,22
2,11,47,70,75,84,32,42,43,34,22,100

There are r-1 (or 11) comparisons in the first pass.

Let X denote the number of comparisons needed by bubble sort and consider $E(X)$ the expected value of X. Then

$$\frac{r(r-1)}{4} \leq E(X) \leq \frac{r(r-1)}{2}.$$

See Ross(2000) for a proof. If $r = 12$, then $33 \leq E(X) \leq 66$.

The quicksort algorithm

Suppose that we want to sort a given set of r distinct values $y_1, y_2, ..., y_r$. A more efficient algorithm than buble sort for doing so is the *quicksort algorithm*, which is recursively defined in Ross(200) p. 55., as follows. When $r = 2$, the algorithm compares the two values and puts them in the appropriate order. When $r > 2$, one of the values is chosen, say it is y_i, and then all of the other values are compared with y_i. Those smaller than y_i are put in a bracket to the left of y_i, and those larger than y_i are put on a bracket to the right of y_i. The algorithm then repeats itself on these brackets, continuing until all values have been sorted. For instance, suppose that we desire to sort the following 10 distinct values:

2,70,11,47,75,100,84,32,42,43,34,22

One of these values is now chosen, say it is 75. We then compare each of the other values to 75, putting those less than 75 in a bracket to the left of 75 and putting those greater than 75 in a bracket to the right of 75. This gives

$\{2, 7, 11, 47, 32, 42, 43, 34, 22\}$, 75, $\{84, 100\}$

We now focus on a bracketed set that contains more than a single value –say the one on the left of the preceding –and choose one of its values –say 11 is chosen. Comparing each of the values in this bracket with 11 and putting the smaller ones in a bracket to the left of 11 and the larger ones in a bracket to the right of 11 gives

$\{2\}$, 11, $\{70, 47, 32, 42, 43, 34, 22\}$, 75 $\{84, 100\}$.

This continues until there is no bracketed set that contains more than a single value.

The Expected number of comparisons is

$$E(X) = 2(r+1)log(r) - 2(r-1).$$

If r= 12, then $E(X) = 42.60$. See Ross(2000) for a proof.