CS 3310 Design and Analysis of Algorithms
Project #2
(Due: 12/6/2022 @Canvas)

Given a list of n numbers, the Selection Problem is to find the k^{th} smallest element in the list. The first algorithm (Algorithm 1) is the most straightforward one, i.e. to sort the list and then return the k^{th} smallest element. It takes O(n log n) amount time. The second algorithm (Algorithm 2) is to apply the procedure Partition used in Quicksort. The procedure partitions an array so that all elements smaller than some pivot item come before it in the array and all elements larger than that pivot item come after it. The slot at which the pivot item is located is called the pivotposition. We can solve the Selection Problem by partitioning until the pivot item is at the k^{th} slot. We do this by recursively partitioning the left subarray if k is less than pivotposition, and by recursively partitioning the right subarray if k is greater than pivotposition. When k = pivotposition, we're done. The best case complexity of this algorithm is O(n) while the worst case is $O(n^2)$. The third algorithm (Algorithm 3) is to apply the Partition algorithm with the mm rule and it's theoretical worst case complexity is O(n).

Write a program to implement the above algorithms for solving the Selection Problem and compare the times. Select-kth 1 is to implement Algorithm 1 using the O(n log n) Mergesort sorting method. Select-kth 2 will implement the Algorithm 2 using the partition procedure of Quicksort iteratively and Select-kth 3 will implement the Algorithm 2 recursively. Select-kth 4 is a recursive procedure to implement the Algorithm 3 with mm rule. Run your program for n = 10, 50, 100, 500, 1000, … (with k = 1, n/4, n/2, 3n/4, and n). In order to obtain more accurate results, the algorithms should be tested with the same list of numbers of different sizes many times. The total time spent is then divided by the number of times the selection is performed to obtain the time taken to solve the given instance. Remember to verify the correctness of your implementation of the four algorithms.

Write a detailed report together with a nicely formatted table to show the average computing times for Select-kth 1, 2, 3, and 4 on random inputs, for n = 10, 50, 100, 500, 1000, … (with k = 1, n/4, n/2, 3n/4, and n). Explain the data sets, test strategies and evaluation of the results. What are the theoretical complexity comparisons of the three algorithms? Find out when (the value of n) does Select-kth 4 become faster than Select-kth 1. Is Select-kth 2 always faster than Select-kth 3? Conclude your report (at least 150 words) with the strength and constraints of your work.

Follow the given Project2-cover-sheet on the class web page for the organization of your report. Turn in your report and your program online @Canvas. Project is due by class time on the due date. Late submission will NOT be accepted.

I encourage discussion among students, but I expect each student to hand in original work. You are responsible for doing your own work and for insuring that your work is protected from copying. The University's policy on Academic Integrity, as stated in the catalog, will be enforced.