Marcus Anderson

Homework 3

CS 6515: Introduction to Graduate Algorithms

My approach:

- To solve this, we will use the fast select algorithm as a black box, while iterating 1 to k – 1 using a loop.
- First, we will need to initialize an empty array to store the number at each quantile. This will be called *quantile_nums*.
- Next, we calculate the partition size that will be multiplied with the k-th quantile by dividing the size of *S*, or *n*, by *k*. This will be called *ps*.
- We then loop through *i = 1 to k* – l and do two things:
    - Call the fast select algorithm with inputs *S*, that represents the array, and *i * ps*, which represents *k*.
    - After the k-th smallest element is found in the fast select algorithm, we add that number to the *quantile_nums* array.
- Once the loop concludes, we return array *quantile_nums*.

Why this works:

The goal of this problem is to find the k-th quantiles of a set of numbers. So, if we have *k* = 4, that means we need to find the numbers in *S* at the 25^{th}, 50^{th}, and 75^{th} percentile. We ignore the number at the 100^{th} percentile since the k-th quantiles of *S* is a subset of exactly k – 1 numbers: $s_1 < s_2 < \ldots s_{k-1}$. This is achieved by calculating the partition size from *n/k* and multiplying it to *i* at each iteration of the 1 to k – 1 loop representing which quantile of *k* to find using the fast select algorithm. The fast select algorithm finds the smallest k-th value and the number is added to our *quantile_nums* array at each iteration until the loop reaches k – l. The fast select algorithm is beneficial to use for this because it analyzes an unsorted set of numbers and adds the k-th quantile numbers to the *quantile_nums* array in a sorted order, which is the expectation of our output.

Runtime:

For this solution, we are using the fast algorithm as a black box, which has a runtime of O(n). However, this algorithm is called k – 1 times in order to get the numbers within *S* at each k-th quantile, which has a runtime of O(k). Finally, both actions of initializing and adding numbers to the *quantile_nums* array both have a runtimes of O(1), making our overall runtime O(nk).

References:

- https://en.wikipedia.org/wiki/Quickselect

Collaborators:

- Lilley, Zachary J: zlilley3@gatech.edu
- Bertrand, James M: jbertrand9@gatech.edu
- Ramasamy, Veerajothi: vramasamy9@gatech.edu
- Acker, Joshua R: jacker7@gatech.edu
- Shah, Jeet Hemant: jshah328@gatech.edu