COP 3502 Suggested Program Edits: Quick Select, Heaps, Bitwise Operators, Hash Tables (Week 10 Programs)

1) Edit the quick select program so that in addition to finding the kth element, if calculates the sum of the array sizes in each of the recursive calls in doing so, which is a very good indicator of the run time. (For example, if we start with an array of size 10 and then after the first partition look at a sub array of size 4, and then after the second partition look for our value in an array of size 1, then the sum of the array sizes is 10 + 4 + 1 = 15.) This value, in the average case, should be a constant times the size of the array. Run this for arrays of size 1,000,000 several times and keep track of the minimum of this sum, the maximum of this sum and the average.

2) Edit the heapexample.c program by creating your own priority rule, different than the one used in the function compare.

3) Add a function to the file bitwise.c where you take in an integer n and return the number of bits set to 1.

4) Add a function to the file bitwise.c where you take in an integer n and return the number of bits the largest value k such that $2^k \leq n$.

5) Pick any of the hash table implementations (hashtable.c, htablelinear.c, hashquadratic.c) and rewrite them to read in all the words in a large dictionary and then search for each one once. Which one of the three performs the best? Does changing the hash function make a difference in performance?