Jessica Sendejo

Final Exam

Question 7.22

7.22 a. For the quicksort implementation in this chapter, what is the running time when all keys are equal?

b. Suppose we change the partitioning strategy so that neither i nor j stops when an element with the same key as the pivot is found. What fixes need to be made in the code to guarantee that quicksort works, and what is the running time when all keys are equal?

c. Suppose we change the partitioning strategy so that i stops at an element with the same key as the pivot, but j does not stop in a similar case. What fixes need to be made in the code to guarantee that quicksort works, and when all keys are equal, what is the running time of quicksort?

Part of code that will need to be changed to make quicksort work:

for( ; ; )

{

while( a[ ++i ] < pivot ) { }

while( pivot < a[ --j ] ) { }

if(i<j)

std::swap( a[ i ], a[j]);

else

break;

}

1. The run time will be O(NlogN)
2. From the code above the while loops would need to be changed so that it doesn’t stop at i or j.

So

while(a[++i] <= pivot && i) {}

while(pivot <= a[--j] && j) {}

The run time would be the worst execution time which is O(N^2)

1. From the code above the while loops would need to be changed so that it stops at i but doesn’t stop at j.

So

// first while loop actually doesn’t need to be changed from the original.

while(a[++i] < pivot) {}

while(pivot <= a[--j] && j > i) {}

The run time would be the worst execution time which is O(N^2)