Adam Rolek

1/30/2020

Exercise 1:

Bool binSearch (int n, array S[], keytype x) {

Index high = n, mid, low = 1;

While (low <= high && location == 0) {

Mid = (low+high)/2;

If (x == S[mid]) {

Return true;

} else if (x < S[mid]) {

High = mid – 1;

} else {

Low = mid + 1;

}

}

}

Exercise 8:

Sequential search has a worst-case time complexity of (n) so it will suffice in many situations. But it would not be very effective in real-time applications, whether they are local or web based. Sequential search would also suffer from situations that require many keys to be searched.

Exercise 10:

The basic operation of this algorithm would be the comparison of the key and the middle values in the array. Assuming we are working with initial array sizes that are multiples of 2, we can count the number of basic operations compared to the arrays size (array size = input size). For an array of 32 items, there would be 6 comparisons. For an array of 64 items, there would be 7 comparisons. This would work out to a time complexity of (lg n + 1)