**Problem 8**

|  |  |
| --- | --- |
| Insertion Sort | Binary Insertion Sort |
| Minimum: 1 comparison  Maximum: n comparison | Minimum: 1 comparison  Maximum: nlog(n) comparison  -Binary Insertion sort (without element shifting):O(nlog n) |

Binary search the position takes O(log N) compares. This makes O(N.log(N)) comparisons for the hole sorting. [We can neglect that N is growing from 1 to the final N while we insert].

The algorithm is still O(n^2) because of the insertions. So, whereas binary search can reduce the clock time (because there are fewer comparisons), it doesn't reduce the asymptotic running time

n \* Binary Search

=n \* O(logn)

=O(nlogn)

n \* (Binary Search + Shifting)

=n \* (O(logn) + O(n))

=O(nlogn) + O(n^2)

=O(n^2)

