CSC108H Winter 2017 Worksheet: Insertion Sort Analysis

1. Insertion Sort: Worst Case

(a) In the list below, 4 passes of the insertion sort algorithm have been completed, and the double bar separates the sorted part of the list from the unsorted part. The item at index $i$ is missing. Fill in the missing item with a value that will cause \texttt{insert(L, i)} to perform the most number of steps. (As a reminder, this is called the \textit{worst case}.)

\[
\begin{array}{cccc|ccc}
L & 3 & 4 & 6 & 6 & 3 & 1 & 5 \\
\end{array}
\]

(b) When \texttt{insert(L, i)} is executed on the example list, how many times does the while loop iterate?

(c) When \texttt{insert(L, i)} is called on the example list, how many \texttt{assignment statements} are executed?

(d) In general, in the \textit{worst} case, on pass $i$ of insertion sort, how many times does the while loop iterate? (Your answer should be a formula that involves $i$.)

(e) In general, in the \textit{worst} case, on pass $i$ of insertion sort, how many \texttt{assignment statements} are executed? (Again, your answer should be a formula that involves $i$.)

(f) In terms of $i$, in the \textit{worst} case, does function \texttt{insert} have constant running time, linear running time, quadratic running time, or some other running time?

\[ \text{(a) constant} \quad \text{(b) linear} \quad \text{(c) quadratic} \quad \text{(d) something else} \]

(g) In function \texttt{insertion_sort}, the first time that function \texttt{insert} is called, $i$ is 0; the second time, $i$ is 1; and so on. What value does $i$ have the last time that function \texttt{insert} is called?

(h) For the call \texttt{insertion_sort(L)}, in the \textit{worst} case, write a formula expressing how many \texttt{comparisons} are made during all the calls to \texttt{insert}.

(i) In the \textit{worst} case, does \texttt{insertion_sort} have constant running time, linear running time, quadratic running time, or some other running time?

\[ \text{(a) constant} \quad \text{(b) linear} \quad \text{(c) quadratic} \quad \text{(d) something else} \]
2. Insertion Sort: Best Case

(a) In the list below, 4 passes of the insertion sort algorithm have been completed, and the double bar separates the sorted part of the list from the unsorted part. The item at index $i$ is missing. Fill in the missing item with a value that will cause $\text{insert}(L, i)$ to perform the fewest number of steps. (That’s called the best case).

\[
\begin{array}{ccccccc}
L & 1 & 3 & 3 & 4 & 8 & 6 & 5
\end{array}
\]

(b) When $\text{insert}(L, i)$ is executed on the example list, how many times does the while loop iterate?

(c) When $\text{insert}(L, i)$ is called on the example list, how many assignment statements are executed?

(d) In general, in the best case, on pass $i$ of insertion sort, how many times does the while loop iterate?

(e) In general, in the best case, on pass $i$ of insertion sort, how many assignment statements are executed?

(f) In the best case, does $\text{insert}$ have constant running time, linear running time, quadratic running time, or some other running time?

\[
\begin{array}{cccc}
\text{(a) constant} & \text{(b) linear} & \text{(c) quadratic} & \text{(d) something else}
\end{array}
\]

(g) For the best case, write a formula expressing how many comparisons are made during all the calls to $\text{insert}$.

(h) In the best case, does $\text{insertion\_sort}$ have constant running time, linear running time, quadratic running time, or some other running time?

\[
\begin{array}{cccc}
\text{(a) constant} & \text{(b) linear} & \text{(c) quadratic} & \text{(d) something else}
\end{array}
\]