Better Lower Bound of average performance of Sorting that sorts by comparisons of keys and removes at most one inversion after each comparison

Recall

Definition

C – a class of sorting algorithms that sort by comparisons of keys and remove at most one inversion after each comparison.

Lemma. While sorting permutaion π and permutation inverse (π) based only on comparisons of keys, each algorithm in class C must perform at least $\frac{1}{2}(n-1)(n+2)$ comparisons.

Proof. In addition to performing a comparison before removing each inversion, the said algorithm must also verify all non – inversions of consecutive elements of the sorted sequence (in order to properly leave them alone). This must take a total of at least $(n-1)$ comparisons in any permutaion π and inverse (π).

So, by virtue of Theorem 2 (in previous .nb file) the number of comparisons is at least

$$\frac{n(n-1)}{2} + n - 1 = \frac{1}{2}(n-1)(n+2)$$

in any permutaion π and inverse (π), or $\frac{1}{4}(n-1)(n+2)$ on average.

This way we proved (Make sure you know why.):

Theorem 3. Each algorithm in class C must perform at least $\frac{1}{4}(n-1)(n+2)$ comparisons on average.

Recall that the upper bound established by Insertion Sort was $\frac{1}{4}n(3+n) - \sum_{j=1}^{n}\frac{1}{j}$.

Thus the difference between that upper bound and the lower bound given by Theorem 3 is

$$\frac{1}{4}n(3+n) - \sum_{j=1}^{n}\frac{1}{j} - \frac{1}{4}(n-1)(n+2) =$$

```
Expand[1/4 n (3 + n) - Sum[1/j, {j, 1, n}] - 1/4 (n - 1) (n + 2)]
```

$$\frac{1}{2} + \frac{n}{2} - \text{HarmonicNumber}[n]$$

$$= \frac{1}{2} + \frac{n}{2} - \sum_{j=1}^{n} \frac{1}{j} \approx \frac{1}{2} + \frac{n}{2} - \text{Log}[n] - 0.577216 =$$

$$\frac{n}{2} - \text{Log}[n] - 0.077216 \approx \frac{n}{2} - \text{Log}[n]$$

The difference between the textbook's lower bound and the upper bound established by Insertion Sort was $\approx n - \text{Log}[n] - 0.577216$

Plot of two lower bounds and the upper bound on the average number of comparisons each algorithm in class C must perform while sorting an array with n elements.

Here is a plot of the textbook lower bound (bottom curve), above lower bound (middle curve), and (an approximation of) the average number of comparisons performed by InsertionSort (top curve). Notice how close the latter two are.

```
Plot[Tooltip[{1/4 (n (n - 1)), 1/4 (n - 1) (n + 2), 1/4 n (n + 3) - Log[n] - 0.577216}], {n, 1, 20}]
```