Insertion Sort

Arvin Kushwaha

November 2021

1 What is Insertion Sort?

Insertion sort is an efficient technique for sorting small sequences. To describe the algorithm, I'll effectively reiterate the description provided in the textbook, ”Introduction to Algorithms, 3e,” which introduces it by means of imagining a set of playing cards (which I will render as just blocks of numbers (representing the associated value: Ace = 1, ..., King = 13), for LaTeX simplicity).

$$\left| \boxed{13}\ \boxed{6}\ \boxed{4}\ \boxed{7}\ \boxed{11}\ \boxed{10}\ \boxed{12}\ \boxed{2}\ \boxed{5}\ \boxed{3}\ \boxed{9}\ \boxed{1}\ \boxed{8} \right|$$

Imagine a set of shuffled diamond cards (pictured above), which we aim to sort from Ace to King. For **insertion sort**, we have a sequence of n values, which we call *keys*. These will be our card values. Now, let us reserve space for our new ordered set of cards.

$$\left| \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \right|$$

$$\left| \boxed{13}\ \boxed{6}\ \boxed{4}\ \boxed{7}\ \boxed{11}\ \boxed{10}\ \boxed{12}\ \boxed{2}\ \boxed{5}\ \boxed{3}\ \boxed{9}\ \boxed{1}\ \boxed{8} \right|$$

To perform **insertion sort**, we take cards one at a time from the unordered sequence and place it in the appropriate location in the to-be ordered sequence. To determine this appropriate location, we compare the card to each card already in the ordered sequence and insert it in the location where our card fails to be greater than the card in the ordered sequence it is being compared to. Let's do this for the first card!

$$\left| \boxed{13} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \right|$$

$$\left| \boxed{6}\ \boxed{4}\ \boxed{7}\ \boxed{11}\ \boxed{10}\ \boxed{12}\ \boxed{2}\ \boxed{5}\ \boxed{3}\ \boxed{9}\ \boxed{1}\ \boxed{8} \qquad \right|$$

Well, that was rather lackluster, right? For the first value, its as simple as simply placing the card in the sequence, since we have no cards to compare the King (13) with.

Let's do the next card: 6. Well, first we have to compare 6 with 13. 6 is less than 13, so it gets placed right at the very beginning. How about 4? Well, comparing 4 to 6, we see that 4 is less than 6 and now 4 is our new first value. Here's our progress so far:

$\mid \boxed{4}\ \boxed{6}\ \boxed{13} \mid$

$\mid \boxed{7}\ \boxed{11}\ \boxed{10}\ \boxed{12}\ \boxed{2}\ \boxed{5}\ \boxed{3}\ \boxed{9}\ \boxed{1}\ \boxed{8} \mid$

Finally, we've come to an interesting card: 7. Let's start the process again. Is 7 greater than 4? Yes? Okay, lets compare with 6. Is 7 greater than 6? Yes? Okay, now for 13. Since 7 is not greater than 13, we place it after 6 and before 13.

$\mid \boxed{4}\ \boxed{6}\ \boxed{7}\ \boxed{13} \mid$

$\mid \boxed{11}\ \boxed{10}\ \boxed{12}\ \boxed{2}\ \boxed{5}\ \boxed{3}\ \boxed{9}\ \boxed{1}\ \boxed{8} \mid$

I think you get the gist of it by now. Lets quickly finish this up:

$\mid \boxed{4}\ \boxed{6}\ \boxed{7}\ \boxed{11}\ \boxed{13} \mid$

$\mid \boxed{4}\ \boxed{6}\ \boxed{7}\ \boxed{10}\ \boxed{11}\ \boxed{13} \mid$

$\mid \boxed{4}\ \boxed{6}\ \boxed{7}\ \boxed{10}\ \boxed{11}\ \boxed{12}\ \boxed{13} \mid$

$\mid \boxed{2}\ \boxed{4}\ \boxed{6}\ \boxed{7}\ \boxed{10}\ \boxed{11}\ \boxed{12}\ \boxed{13} \mid$

$\mid \boxed{2}\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{10}\ \boxed{11}\ \boxed{12}\ \boxed{13} \mid$

$\mid \boxed{2}\ \boxed{3}\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{10}\ \boxed{11}\ \boxed{12}\ \boxed{13} \mid$

$\mid \boxed{2}\ \boxed{3}\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{9}\ \boxed{10}\ \boxed{11}\ \boxed{12}\ \boxed{13} \mid$

$\mid \boxed{1}\ \boxed{2}\ \boxed{3}\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{9}\ \boxed{10}\ \boxed{11}\ \boxed{12}\ \boxed{13} \mid$

$\mid \boxed{1}\ \boxed{2}\ \boxed{3}\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{8}\ \boxed{9}\ \boxed{10}\ \boxed{11}\ \boxed{12}\ \boxed{13} \mid$

Just by following two simple steps, we've constructed a sorted sequence from an unordered sequence. Remember at the beginning, when I said this technique is efficient for sorting small sequences? Why is that?

2 Analyzing Insertion Sort

For this section, I'm going to assume some familiarity with big-O notation. If not, no worries, but I highly recommend taking a look at the document about O-notation in the root of the `implementation/` directory in the repository.