Sorting Algorithms

Friday, 8 September 2023 10:23 AM

Bubble Sort

https://www.programiz.com/dsa/bubble-sort

Bubble sort is a simple sorting algorithm that repeatedly steps through the array to be sorted. It compares adjacent items (pairs of adjacent array elements) and exchanges them if they are not in the correct order (ascending or descending). The algorithm makes multiple passes until no swaps are necessary and the elements of the array are sorted. The algorithm is named for the way elements "bubble" to the top of the array. After each loop, one less element (the leftmost) needs to be compared. The algorithm is very slow and impractical for most cases.

Programming Example 21: Bubble sort

```
//==== Bubble Sort ====
ELEMENTS = [1,663,8,2,4,1,22,66,20,122]
loop I from 0 to 8// Number of elements - 2
    loop J from 0 to (8-I)
        if ELEMENTS[J] < ELEMENTS[J + 1] then //for descending order
OR
        if ELEMENTS[J] > ELEMENTS[J + 1] then //for ascending order
           TEMP = ELEMENTS[J]
           ELEMENTS[J] = ELEMENTS[J+1]
           ELEMENTS[J+1] = TEMP
        end if
    end loop
end loop

output "Sorted elements"

loop E from 0 to 9
   output ELEMENTS[E]
end loop
```

Selection Sort:

https://www.programiz.com/dsa/selection-sort

Selection sort is a very simple and inefficient sorting algorithm that divides the input array into two sub-arrays: the first sub-array contains the already sorted elements, and the second

sub-array contains the unsorted elements and occupies the rest of the array. The first sub-arrays is built up from left to right at the lowest index position [I=0]. At the beginning, the sub-array that contains the sorted elements is empty and the sub-array that contains the unsorted element is the entire array. The algorithm continues by finding the smallest (or largest, depending on the sorting order) element in the sub-array that contains the unsorted elements, exchanging it with the leftmost unsorted element (element located in the lowest index position) and putting it in sorted order. The algorithm then moves the first sub-array borders one element to the right.

Programming Example 23: Selection sort

```
//==== Selection Sort ====

ELEMENTS = [1,5,3,86,256,420,9,510,51,24,60]
MIN = 0
I = 0
TEMP = 0

loop MIN from 0 to 9
    I = MIN
    loop CURRENT from MIN+1 to 10
        if ELEMENTS[CURRENT] > ELEMENTS[I] then//for descending order
OR
        if ELEMENTS[CURRENT] < ELEMENTS[I] then//for ascending order
            I = CURRENT
        end if
    end loop
    TEMP = ELEMENTS[I]
    ELEMENTS[I] = ELEMENTS[MIN]
    ELEMENTS[MIN] = TEMP
end loop

output "SORTED ARRAY"
loop C from 0 to 10
   output ELEMENTS[C]
end loop
```