Sorting techniques

Algorithms:

An algorithm is a method of representing the step-by-step procedure for solving a problem.

Sorting:

Arranging the elements in a list either in ascending or descending order. Various sorting algorithms are:

1) Bubble sort
2) Selection sort
3) Insertion sort
4) Quick sort
5) Merge sort
6) Heap sort

curious_programmer

Bubble Sort

The bubble sort is an example of exchange sort. In this method, repetive comparison is performed among elements and essential swapping of element is done. Bubble sort is commonly used in sorting algorithm. It is easy to understand but time consuming it takes more number of comparison to sort a list. In this type two successive elements are compared and swapping

is done. Thus, step-by-step entire array elements are cheked. It is different from the selection sort. Instead of searching the minimum element and then applying swapping, two records are swapped instantly upon noticing that they are not in order.

Algorithm:

```
Bubble_Sort (A[ ], N)
 step 1 : start
 step2 : Take an array of n elements
 step3 : for i=0, ..... n-2
 step4 : for j=i+1, ..... n-1
 step5 : if arr[j] > arr[j+1] then
           Interchange arr[i] and arr[j+1]
         End of if
 step6 : Print the sorted array arr
 step7 : stop
```

curious_programmer

Selection Sort

Selection sort (select the smallest and Exchange) The first item is compared with the remaining n-1 items, and whichever of all is lowest, is put in the first position. Then the second item from the list is taken and compared with the remaining (n-2) items, if an item with a value less than that of the second item is found on the (n-2) items, it is swapped (Interchanged) with the second item of the list and so on.

selection sort	comparisons
[8] 5 7 [1] 9 3	(n-1) first smallest
1 [5] 7 8 9 [3]	(n-2) second smallest
1 3 [7] 8 9 [5]	(n-3) third smallest
1 3 5 [8] 9 [7]	2
1 3 5 7 [9] [8]	1
1 3 5 7 8 9	0

Insertion Sort

Insertion sort iterates, consuming one input element each repetition, and growing a sorted output list. Each iteration, insertion sort removes one element the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remains.

Algorithm: curious-programmer

step1 : start
step2 : For i ← 1 to length(A)
step3 : j ← i
step4 : while j>0 and A[j-1] >A[j]
step5 : swap A[j] and A[j-1]

Step6 : j ← j-1
Step7 : end while
Step8 : end for
Step9 : stop

Quick Sort

Quick sort is a divide and conquer algorithm. Developed by Tony Hoare in 1959.
Quick sort first divides a large array into two smaller sub-arrays : the low elements and the high elements.
Quick sort can then recursively sort the sub-arrays.

Algorithm : Curious-programmer

Step1 : Pick an element, called a pivot, from the array
step2 : Partitioning :
reorder the array so that all elements with values less than the pivot come before the pivot, while all elements with values greater than the pivot come after it (equal values can go either way). After this partitioning, the pivot is in its final position. This is called the partition operation.
step3 : Recursively apply the above steps to the sub-array of elements with similar smaller values and separately to the sub-array of elements with greater values

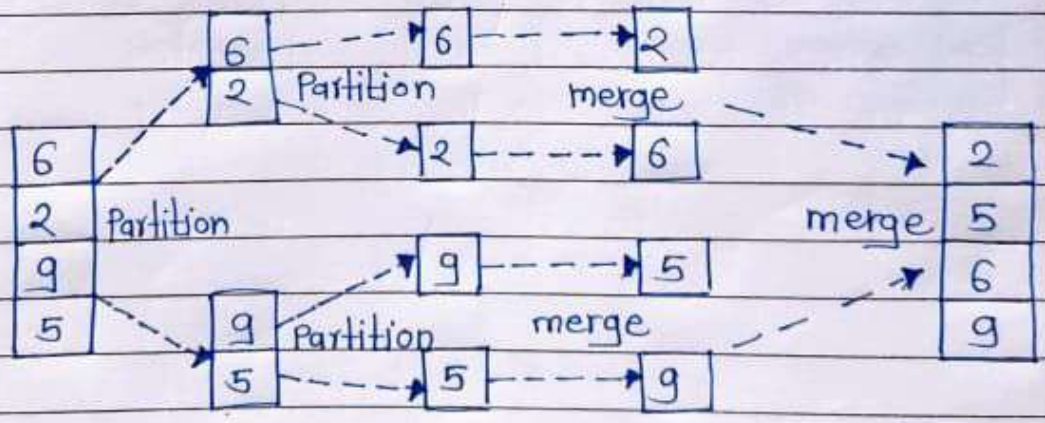
Merge Sort

Merge sort is a sorting technique based on divide and conquer technique. In merge sort the unsorted list is divided into N sublists, each having one element, because a list of one element is considered sorted. Then, it repeatedly merge these sublists, to produce new sorted sublists, and at lasts one sorted list is produced. Merge sort is quite fast, and has a time complexity of $O(n \log n)$.

Conceptually, merge sort works as follows:

curious - programmer

1. Divide the unsorted list into two sub-lists of about half the size.
2. Divide each of the two sub lists recursively until we have list sizes of length 1, in which case the list itself is returned.
3. Merge the two sub lists back into one sorted list.



The result is : 2 5 6 9

Heap Sort

Heap sort is a completely binary tree with the property that a parent is always greater than or equal to either of its children (if they exist). First the heap (max or min) is created using binary tree and then heap is sorted using priority queue.

Steps:

step1: start with just one element.
step2: Insert next elements and make this heap.
step3: Repeat step 2, untill the all elements are included in the heap.

curious_.programmer

Date :

@Curious_.programmer

PDF uploaded on
Telegram
(Link in bio)

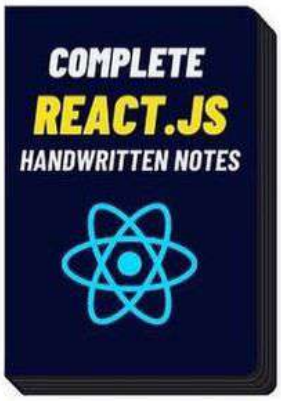
PYTHON, C,
C++, Java

handwritten notes
Available on Telegram

Search Telegram :- " curious coder "

@curious_.programmer

Buy This Complete Premium Notes @ Rs. 149



LINK IN BIO