7th Dec 2010

Searching and sorting

Overview of various algos

(i) Bubble sort

Time Complexity

: Best $O(n)$:- Inner loop will never run (array is already sorted)

: Worst $O(n^2)$: - Array is reverse sorted

Space Complexity : $O(1)$

Stability : stable

In place :? __True__

Uses (i) Array is of small size
(ii) Array is of large size but nearly sorted.

Stability :- A sorting algorithm is said to be stable if two objects with equal keys appear in same order in sorted output as they appear in input array to be sorted.

In-place : An in-place algorithm is one that does not need extra space and produces an output in the same memory that contains the data by transforming input 'in-place'. However a small constant extra space is allowed.

(ii) Selection sort

Time Complexity : Best & Worst & Average $\Rightarrow O(n^2)$

Space Complexity : $O(1)$
Stability : Not stable
Inplace ? : True

022544

Uses: (i) Array is of small size
(ii) To minimize no. of swaps

Remarks: Bubble sort has more number of swaps as compared to selection sort. Bubble sort can also be implemented stably.

Selection sort makes $O(n)$ swaps which is minimum among all sorting algorithms mentioned above.

(iii) Insertion Sort

Time Complexity : Best $O(n)$
Worst : $O(n^2)$
Space Complexity $O(1)$
Stability : Stable
In place : true

Uses:
(i) Array is of small size
(ii) Array is nearly sorted

Remark:- Standard library of C uses this also when n becomes smaller than a threshold. For small size it is better than merge or quick sort, because of low constant values and non-recursive nature.

(iv) Counting Sort

    Time Complexity : Best & worst & Average $\Rightarrow$ O(n+K)

    Space Complexity : O(n+K)

    stability : not - stable

    In place : true

(v) Merge Sort

    Time Complexity : Best O(n log n)

                    worst O(n²)

    Space Complexity O(n)

    Stability : stable

    In place :- false

    When to use (i) We don't have random
                            access (linked list)

              (ii) When array is not too large.

(vi) Quick sort

    Time complexity $\Rightarrow$ O(n log n)

                  worst O(n²)

    Space complexity : O(n)

    stability : Not - stable

    In place : true

Uses (i) preferred over merge sort for extremely
        long arrays

    (ii) When you don't care about worst case
        complexity.