summary page

- **The five sorting algorimthms are :**
- Cocktail , Comb, Counting, Gnome, Strand

- **Stable Definition :simply** The relative order of items with equal keys is preserved

➔ Cocktail (**Stable** & comparison sort & inplace)

Data structure	Array
Worst case performance	$O(n^2)$
Best case performance	$O(n)$
Average case performance	$O(n^2)$
Worst case space complexity	$O(1)$

An example of a list that proves the need for cocktail sort :
A list (2,3,4,5,1), which would only need to go through one pass of cocktail sort to become sorted, but if using an ascendin gbubble sort would take four passes.

➔ Comb (inplace ,not stable)

Data structure	Array
Worst case performance	$O(n^2)$[1]
Best case performance	$O(n)$
Average case performance	$\Omega(n^2/2^p)$, where p is the number of increments[1]
Worst case space complexity	$O(1)$

The basic idea is to eliminate small values near the end of the list,using a gap ..

Gap of initiatly [input size / shrink factor] shrink factor = 1.3 ,,after testing over 200,000 random lists

==

➔ Counting (Stable,only for non negative number)

used as a subroutine in another sorting algorithm, radix sort,

➔takes linear time :

Time Complexity Analysis

- So the counting sort takes a total time of: $O(n + k)$
- Counting sort is called stable sort.
  - A sorting algorithm is ***stable*** when numbers with the same values appear in the output array in the same order as they do in the input array.

➔ Gnome(inplace & stable)

moving an element to its proper by a series of swaps

Data structure	Array
Worst case performance	$O(n^2)$
Best case performance	$O(n)$
Average case performance	$O(n^2)$
Worst case space complexity	$O(1)$ auxiliary

➔Strand (not stable,not in place)

The Idea is pulling sorted sublists out of the list to be sorted and merging them with a result array

Data structure	Linked list
Worst case performance	$O(n^2)$
Best case performance	$O(n)$
Average case performance	$O(n^2)$
Worst case space complexity	$O(1)$ auxiliary