HACETTEPE UNIVERSITY

BBM204 SOFTWARE PRACTICUM

ASSIGNMENT 1

MUSTAFA KOLLU 21627485

AIM:

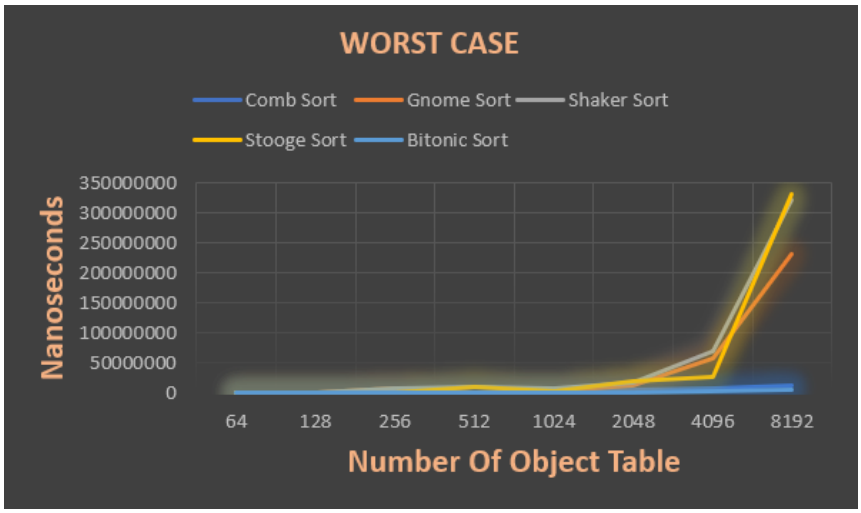
In this experiment, I will analyse different algorithms and compare their run times. I will measure the working times of 5 different sorting algorithms and interpret the results. My aim here is to show the relationship between the running time of applications and theoretical asymptotic complexities.

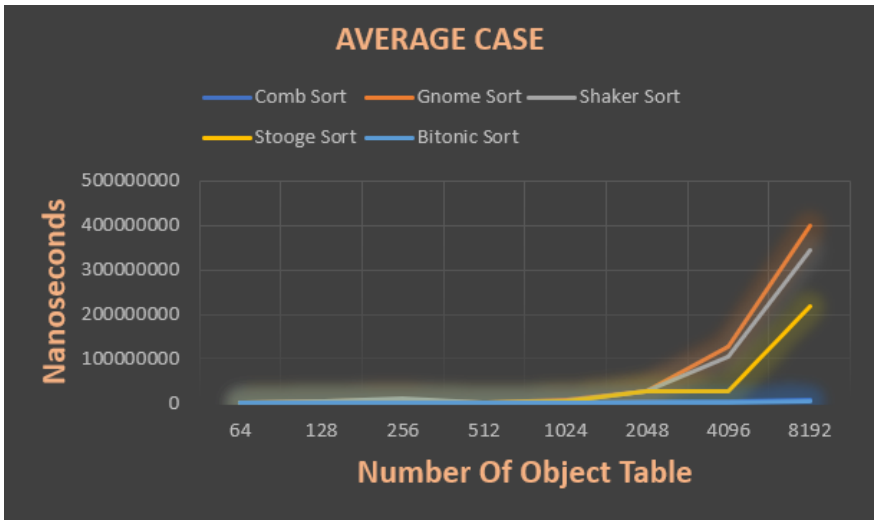
Data:

Sort algorithms execution time (Nanosecond) in Java (IDE = Eclipse)

Algorithms/n	64	128	256	512	1024	2048	4096	8192
Comb Sort	72400	228000	634100	1244900	4334400	5996800	8159900	12689000
Gnome Sort	1145900	1879800	7847000	1440500	4556100	12890300	58723600	232336600
Shaker Sort	923200	2287500	7477800	10862300	8158200	16582600	69629300	320592900
Stooge Sort	244280	830880	1186720	10342490	3282352	19568434	27034954	331688822
Bitonic Sort	89300	705500	450000	212200	782800	1373600	3232100	5996800
			Worst Case					

Algorithms/n	64	128	256	512	1024	2048	4096	8192
Comb Sort	118900	172900	370100	1228000	1923100	3022600	4923300	6089200
Gnome Sort	1263600	5129800	6368400	1382500	8331700	26061400	128581800	400162000
Shaker Sort	953400	3930600	11522600	2526300	5312400	28367500	103534300	345371200
Stooge Sort	457290	210410	1073280	1035276	3393834	28256681	26538681	219217758
Bitonic Sort	120460	171200	87200	159300	575000	788100	1855800	3850900
			Average Case					





Algorithms Stability:

Stable sort algorithms sort repeated elements in the same order that they appear in the input. When sorting some kinds of data, only part of the data is examined when determining the sort order. For example, in the card sorting example to the right, the cards are being sorted by their rank, and their suit is being ignored. This allows the possibility of multiple different correctly sorted versions of the original list. Stable sorting algorithms choose one of these, according to the following rule: if two items compare as equal, like the two 5 cards, then their relative order will be preserved, so that if one came before the other in the input, it will also come before the other in the output.

Stable Sorting Algorithms:

- Cocktail Sort
- Gnome Sort

Unstable Sorting Algorithms:

- Comb Sort
- Bitonic Sort
- Stooge Sort

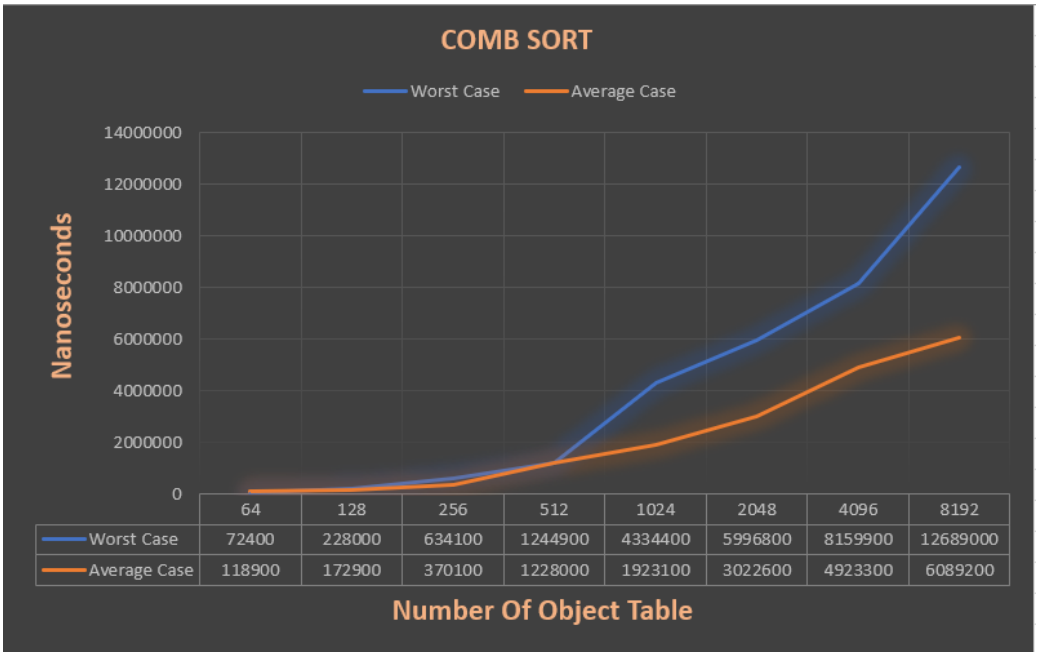
Algorithms:

1. **Comb Sort**

   The basic idea of comb sort and the bubble sort is same. In other words, comb sort is an improvement on the bubble sort. In the bubble sorting technique, the items are compared with the next item in each phase. But for the comb sort, the items are sorted in a specific gap. After completing each phase, the gap is decreased. The decreasing factor or the shrink factor for this sort is 1.3. It means that after completing each phase the gap is divided by 1.3.

   Complexity

   Average time complexity - $O(n^2/2^p)$
   Best-Case time complexity - O(n log n)
   Worst-Case time complexity - $O(n^2)$



2. **Gnome Sort**

The gnome sort is a sorting algorithm which is similar to insertion sort in that it works with one item at a time but gets the item to the proper place by a series of swaps, similar to a bubble sort. It is conceptually simple, requiring no nested loops. The average running time is O(n^2) but tends towards O(n) if the list is initially almost sorted. The algorithm finds the first place where two adjacent elements are in the wrong order and swaps them. It takes advantage of the fact that performing a swap can introduce a new out-of-order adjacent pair next to the previously swapped elements. It does not assume that elements forward of the current position are sorted, so it only needs to check the position directly previous to the swapped elements.

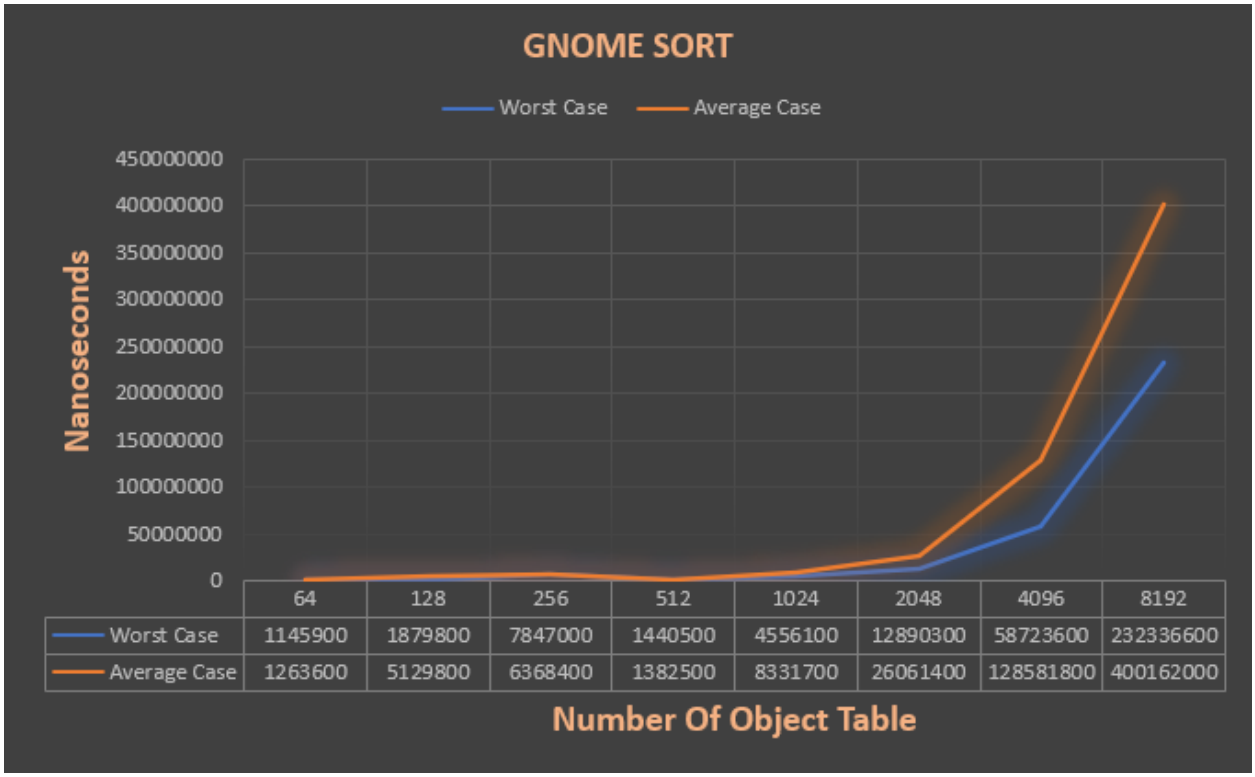
Complexity

As there are no nested loop (only one while) it may seem that this is a linear O(N) time algorithm. But the time complexity is O(N^2). This is because the variable – 'index' in our program doesn't always gets incremented, it gets decremented too. However, this sorting algorithm is adaptive and performs better if the array is already/partially sorted.

Average time complexity - $O(n^2)$

Best-Case time complexity - O(n)

Worst-Case time complexity - $O(n^2)$



	64	128	256	512	1024	2048	4096	8192
Worst Case	1145900	1879800	7847000	1440500	4556100	12890300	58723600	232336600
Average Case	1263600	5129800	6368400	1382500	8331700	26061400	128581800	400162000

3. **Cocktail Sort**

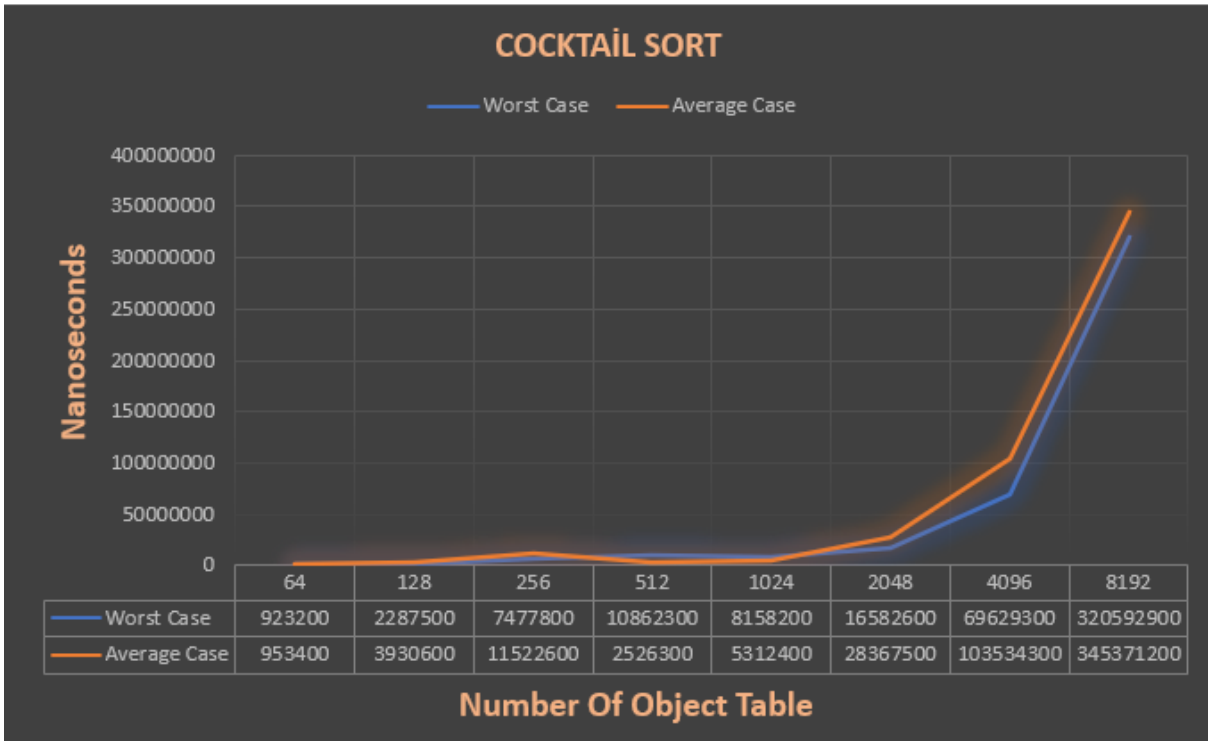
Cocktail sort, also known as bidirectional bubble sort, cocktail sort, shaker sort (which can also refer to a variant of selection sort), ripple sort, shuffle sort, or shuttle sort, is an extension of bubble sort. The algorithm extends bubble sort by operating in two

directions. While it improves on bubble sort by more quickly moving items to the beginning of the list, it provides only marginal performance improvements. Like most variants of bubble sort, cocktail shaker sort is used primarily as an educational tool. More performant algorithms such as timsort, or merge sort are used by the sorting libraries built into popular programming languages such as Python and Java.

Complexity

The complexity of the cocktail shaker sort in big O notation O(n^2) is for both the worst case and the average case, but it becomes closer to if the list is mostly ordered before applying the sorting algorithm. For example, if every element is at a position that differs by at most k ($k \geq 1$) from the position it is going to end up in, the complexity of cocktail sort becomes O(kn).

Average time complexity - $O(n^2)$
Best-Case time complexity - O(n)
Worst-Case time complexity - $O(n^2)$



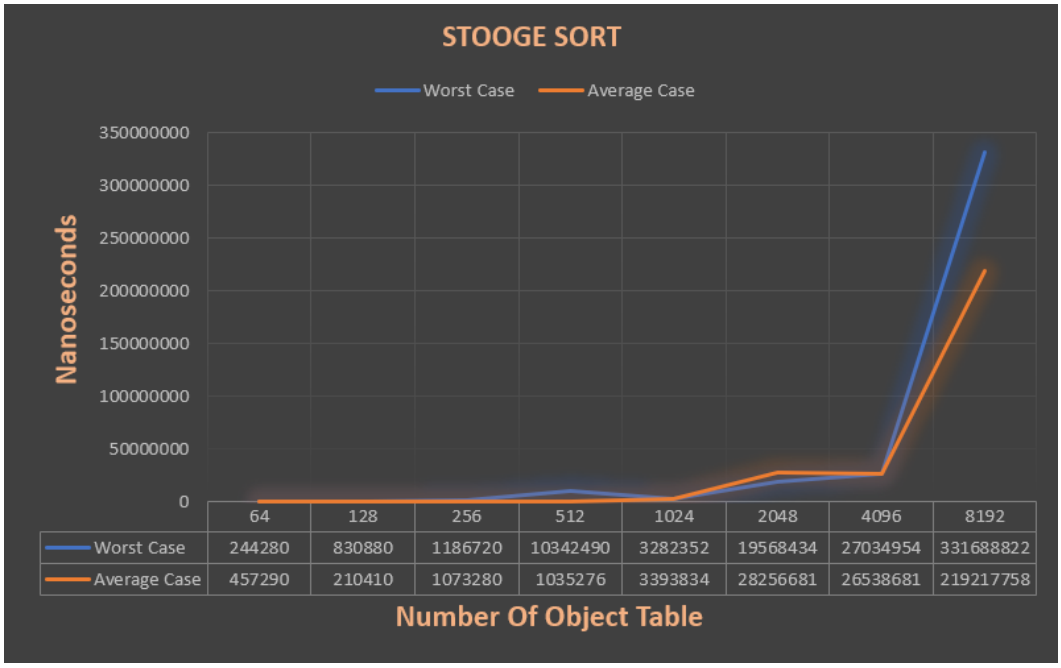
	64	128	256	512	1024	2048	4096	8192
Worst Case	923200	2287500	7477800	10862300	8158200	16582600	69629300	320592900
Average Case	953400	3930600	11522600	2526300	5312400	28367500	103534300	345371200

4. **Stooge Sort**

Stooge Sort is a recursive sorting algorithm. It is inefficient but interesting sorting algorithm. It divides the array into two overlapping parts (2/3 each). Then it performs sorting in first 2/3 part and then it performs sorting in last 2/3 part. After that, sorting is done on first 2/3 part to ensure the array is sorted.

Complexity

Average time complexity - O(n ^ log(3) /log(1.5))
Best-Case time complexity - O(n ^ log(3) /log(1.5))
Worst-Case time complexity - O(n ^ log(3) /log(1.5))



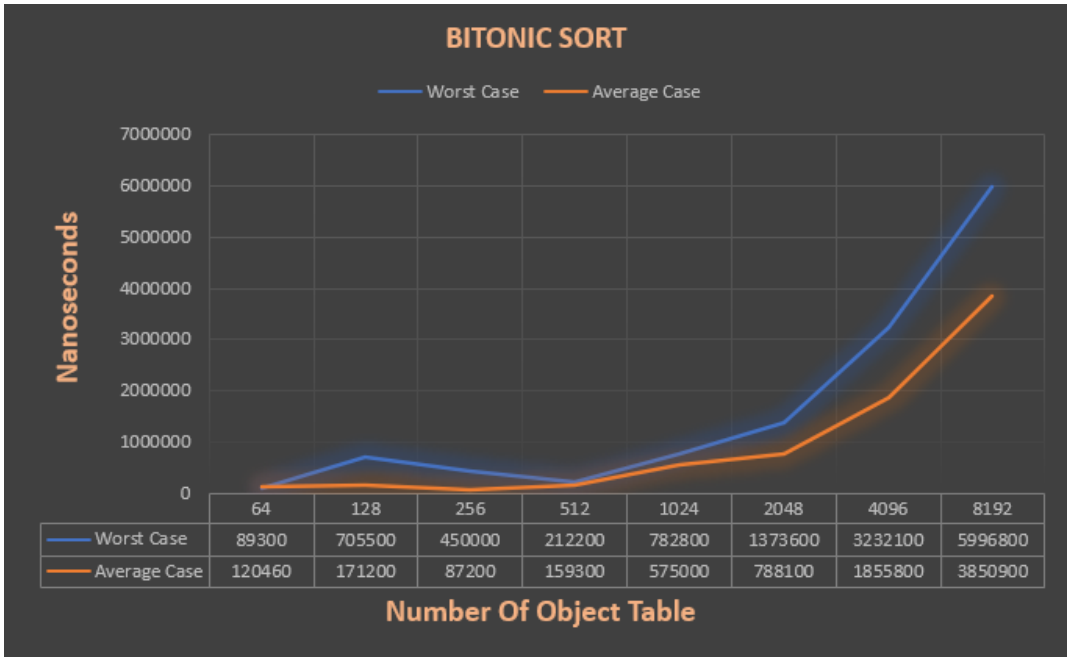
	64	128	256	512	1024	2048	4096	8192
Worst Case	244280	830880	1186720	10342490	3282352	19568434	27034954	331688822
Average Case	457290	210410	1073280	1035276	3393834	28256681	26538681	219217758

5. Bitonic Sort

Bitonic is a parallel algorithm for sorting. It is also used as a construction method for building a sorting network. The algorithm was devised by Ken Batcher. The resulting sorting networks consist of O(nlog^2(n)) comparators and have a delay of O(log^2(n)), where n is the number of items to be sorted. A sorted sequence is a monotonically non-decreasing (or nonincreasing) sequence. A bitonic sequence is a sequence with for some, or a circular shift of such a sequence.

Complexity

Average time complexity - $O(\log^2 n)$
Best-Case time complexity - $O(\log^2 n)$
Worst-Case time complexity - $O(\log^2 n)$



	64	128	256	512	1024	2048	4096	8192
Worst Case	89300	705500	450000	212200	782800	1373600	3232100	5996800
Average Case	120460	171200	87200	159300	575000	788100	1855800	3850900