

Algorithm's Repository

20358446

Analysis

Details regarding Implementations of Algorithms are present in Java Documents above each Algorithm.

-----Sorting Algorithms-----

Bogo Sort

Time Complexity

Best Case	Average	Worst Case
$O(n)$	$O(n*n!)$	$O(\infty)$

Space Complexity : $O(1)$

Merge Sort

Time Complexity

Best Case	Average	Worst Case
$O(n\log(n))$	$O(n\log(n))$	$O(n\log(n))$

Space Complexity : $O(n)$

Insertion Sort

Time Complexity

Best Case	Average	Worst Case
$O(N)$	$O(N^2)$	$O(N^2)$

Space Complexity : $O(1)$

Selection Sort

Time Complexity

Best Case	Average	Worst Case
$O(N^2)$	$O(N^2)$	$O(N^2)$

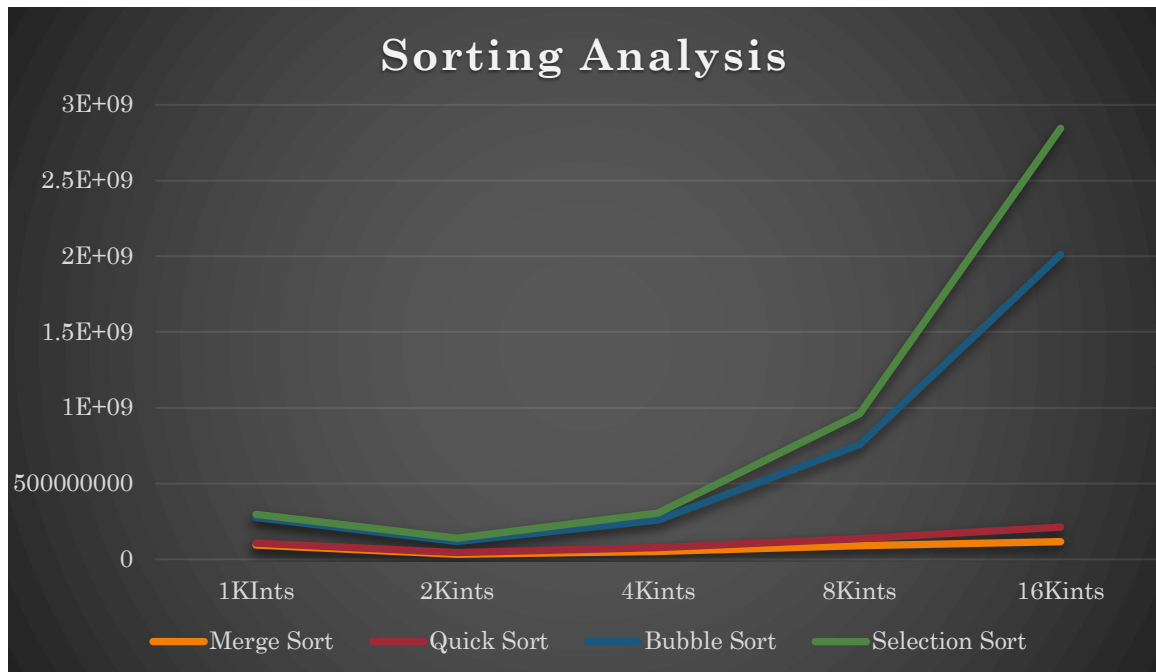
Space Complexity : $O(1)$

Quick Sort

Time Complexity

Best Case	Average	Worst Case
$O(n\log(n))$	$O(n\log(n))$	$O(N^2)$

Space Complexity : $O(\log(n))$



```
-----Sorting Analysis-----
-----Merge Sort-----
Time Taken for File 1 = 94720011 ns
Time Taken for File 2 = 33067358 ns
Time Taken for File 3 = 53029603 ns
Time Taken for File 4 = 89079949 ns
Time Taken for File 5 = 116471769 ns
Total Time taken for all 5 Files = 386368690 ns
-----Quick Sort-----
Time Taken for File 0 = 10243311 ns
Time Taken for File 1 = 11533932 ns
Time Taken for File 2 = 22658438 ns
Time Taken for File 3 = 46222326 ns
Time Taken for File 4 = 96045023 ns
Total Time taken for all 5 Files = 573071720 ns
-----Bubble Sort-----
Time Taken for File 0 = 171401863 ns
Time Taken for File 1 = 72621862 ns
Time Taken for File 2 = 182785628 ns
Time Taken for File 3 = 620362655 ns
Time Taken for File 4 = 1799707534 ns
Total Time taken for all 5 Files = 3419951262 ns
-----Selection Sort-----
Time Taken for File 0 = 21475780 ns
Time Taken for File 1 = 22394377 ns
```

Time Taken for File 2 = 47347402 ns
Time Taken for File 3 = 205298754 ns
Time Taken for File 4 = 833313162 ns
Total Time taken for all 5 Files = 4549780737 ns

-----Compression Algorithms-----

- Run Length Encoding

Huffman Compression

File:	Uncompressed Bits	Compressed Bits	Compression Ratio
mobydick.txt	5145680	2921328	1.76142
medTale.txt	45056	23912	1.88424
genomeVirus.txt	50008	12576	3.97646
q32x48.bin	1536	816	1.88235

We see that Huffman Compression is a lossless Compression algorithm

File:	Uncompressed Bits	Compressed Bits	After Uncompressing
mobydick.txt	5145680	2921328	5145680 bits
medTale.txt	45056	23912	45056 bits
genomeVirus.txt	50008	12576	50008 bits
q32x48.bin	1536	816	1536 bits

Huffman VS RunLength

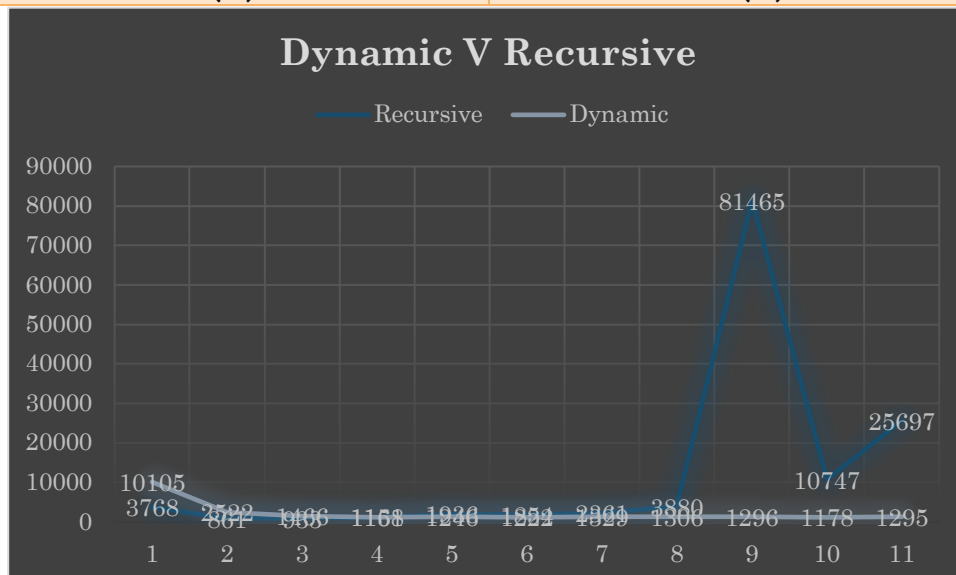
Q34x48.bin	Huffman	RunLength
Compressed bits	816	1114

After some research, it appears that difference can be attributed to RunLength compressing better by Exploiting long runs of repeated characters.

-----Dynamic Programming-----

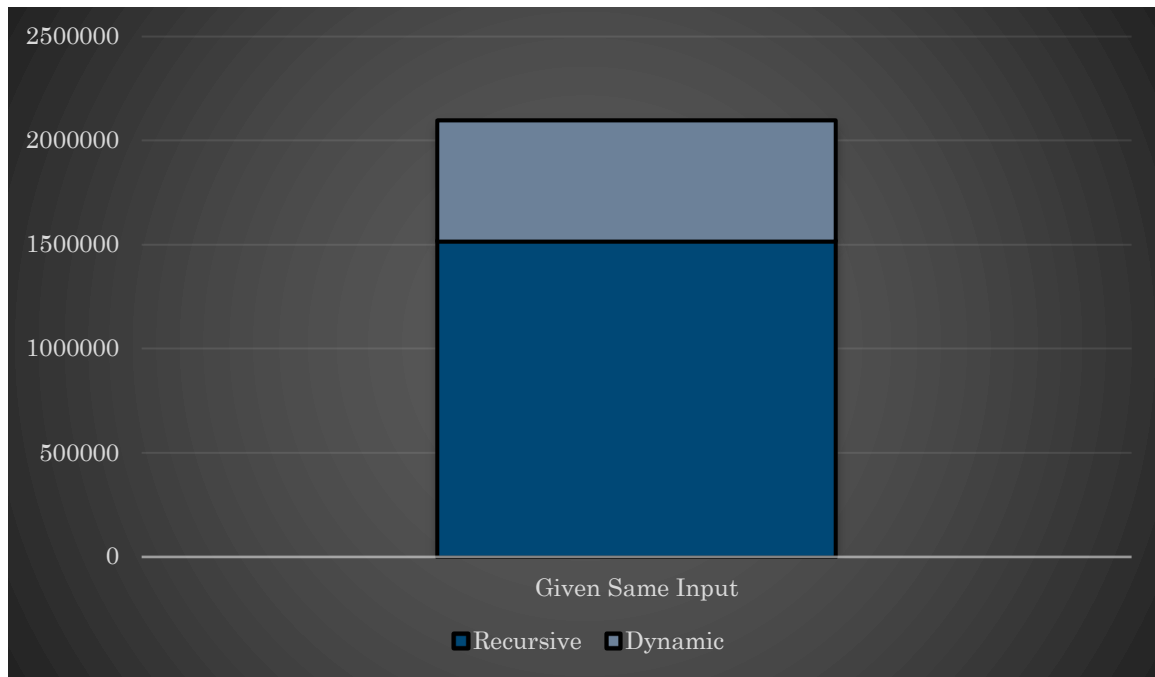
Fibonacci

Time Complexity	
BruteForce	Dynamic Programming
$O(2^n)$	$O(n)$
Space Complexity	
BruteForce	Dynamic Programming
$O(n)$	$O(1)$



Knapsack

Time Complexity	
BruteForce	Dynamic Programming
$O(2^n)$	$O(N * CAPACITY)$
Space Complexity	
BruteForce	Dynamic Programming
$O(N * CAPACITY)$	$O(N * CAPACITY)$



-----Knapsack Recursive-----

Time Taken for Knapsack Recursive = 1517769 ns

-----Knapsack Dynamic-----

Time Taken for Knapsack Dynamic = 573187 ns

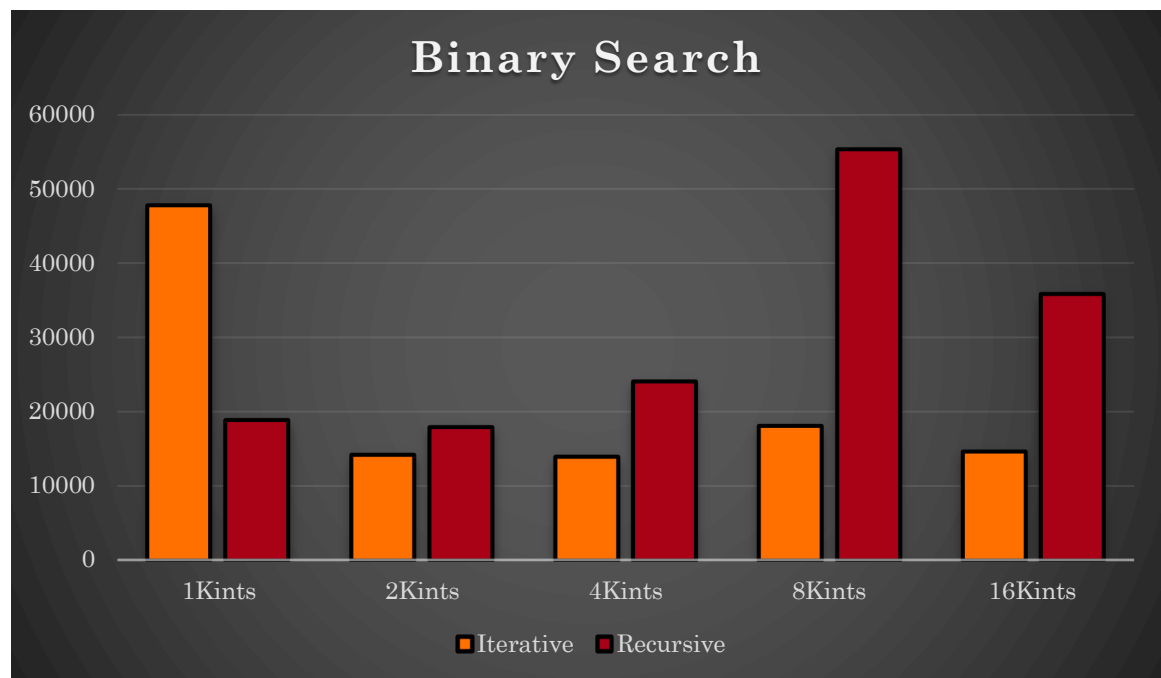
Longest Common Substring

Time Complexity	
$O(n*m)$	
Space Complexity	
$O(n)$	

-----Searching Algorithms-----

Binary Search

Time Complexity	
Iterative	Recursive
$O(\log(n))$	$O(\log(n))$
Space Complexity	
$O(1)$	$O(\log(n))$



-----Binary Search Analysis-----

-----Binary Search Non Recursive-----

Time Taken for File 1 = 47720 ns

Time Taken for File 2 = 14201 ns

Time Taken for File 3 = 13908 ns

Time Taken for File 4 = 18116 ns

Time Taken for File 5 = 14664 ns

-----Binary Search Recursive-----

Time Taken for File 6 = 18882 ns

Time Taken for File 7 = 17889 ns

Time Taken for File 8 = 24032 ns

Time Taken for File 9 = 55187 ns

Time Taken for File 10 = 35754 ns

String Search

Time Complexity

$O(n*m)$

KMP Search

Time Complexity
$O(n+m)$
Space Complexity
$O(m)$

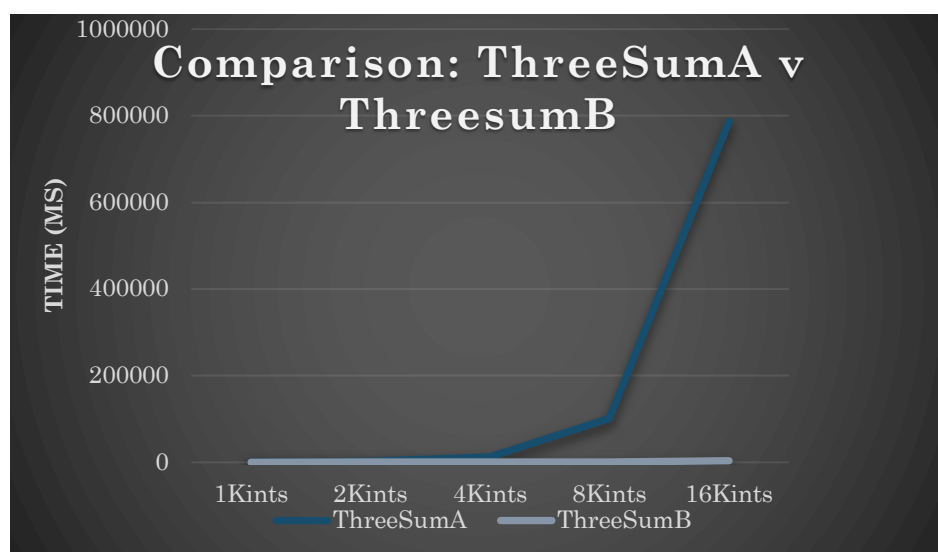
-----Miscellaneous-----

ThreeSumA

Time Complexity
$O(n^3)$
Space Complexity
$O(n)$

ThreeSumB

Time Complexity
$O(n^2)$
Space Complexity
$O(n)$



	ThreeSumA	ThreeSumB
1Kints	240	22
2Kints	1611	53

4Kints	11864	179
8Kints	100852	944
16Kints	786638	3642

Russian Peasant Multiplication

Time Complexity
$O(1)$