	algorithm effic	iency											
Section 1 : number of compari	icone & ewane, an	nd time taken to s	ort an uncorted/cor	tod array on 3 o	lifforont algorithms								
features /algorithm	isonis & swaps, an	size of 10	ort ari urisorteu/sor	ted allay oil 5 c	illierent algoritims	•	size of 100				size of 1000		
array size	shellSort	bubbleSort	bubbleSort2	min value		shellSort	bubbleSort	bubbleSort2	min value	shellSort	bubbleSort	bubbleSort2	min value
array size	SHEIIOUT	bubbleoort	DubbleGortz	miin value		SHEIIOUT	bubbleoort	bubblesonz	min value	SHEIIOOIT	bubblecort	DubbleGortz	min value
UNSORTED													
number of comparisons	53	45	45	45		2808	4950	4905	2808	55727	499500	499065	5572
number of swaps made	12	30	30	12		395	2382	2382	395	7455	260612	260612	745
total time taken to sort (ME)	0.0208	0.0078	0.0076	0.0076		0.1978	0.3185	0.9358	0.1978	4.5337	27.3134	24.49	4.533
				0					0				
SORTED				0					0				
number of comparisons	22	45	9	9		503	4950	99	99	8006		999	99
number of swaps made	0	0	0	0		0	0		0	0			
total time taken to sort	0.0036	0.0051	0.0019	0.0019		0.0893	0.184	0.0067	0.0067	0.9166	13.2808	0.0939	0.093
Section 2 : An emphasis algor	ithm officionay on	3 different algorit	hme										
Coulon 2 . An emphasis algui	•	time taken (ME)											
input size	shellSort	bubbleSort	bubbleSort2	Grap	n 1 : Comparisor	of typical 3 diffe	rent algorithms	on unsorted	Graph 2 : Comp	parison of typical 3 different a	algorithms on so	rted array	
10	0.0208	0.0078	0.0076			array			_	shellSort - bubbleSort -	bubbleSort2		
100	0.1978	0.3185	0.9358		shellSo	rt - bubbleSort	bubbleSort2	2	15				
1000	4.5337	27.3134	24.49	3	0								
									y 10				
				₩ 2	0				_ ≥ 10				
					0			_	-				
	1	time taken (ME)		(eu (akeu				
input size	shellSort	time taken (ME) bubbleSort	bubbleSort2	e taken (0				ne taken				
input size		` ,	bubbleSort2 0.0019	Time taken (ME)					Time taken (ME)				
	shellSort	bubbleSort		Time taken (0				Time taken				
10	shellSort 0.0036	bubbleSort 0.0051	0.0019	Time taken (400	600	800 1000	Time taken	200 400	600 800	1000	
10 100	shellSort 0.0036 0.0893	bubbleSort 0.0051 0.184	0.0019 0.0067	Time taken (0	400 input		800 1000	Тше гакен Тше гакен Тше гакен Тше гакен	200 400 input size		1000	
10 100	shellSort 0.0036 0.0893	bubbleSort 0.0051 0.184	0.0019 0.0067	Time taken (0			800 1000	Time taken			1000	
10 100	shellSort 0.0036 0.0893 0.9166	bubbleSort 0.0051 0.184 13.2808	0.0019 0.0067	Time taken (0			800 1000	Time taken			1000	
10 100 1000	shellSort 0.0036 0.0893 0.9166	bubbleSort 0.0051 0.184 13.2808	0.0019 0.0067	Time taken (0			800 1000	Time taken			1000	
10 100 1000 Section 3 : Theoretical discuss	shellSort 0.0036 0.0893 0.9166	bubbleSort 0.0051 0.184 13.2808	0.0019 0.0067 0.0939		0 200	input		800 1000	Time taken			1000	
10 100 1000 Section 3 : Theoretical discuss Discussion :	shellSort 0.0036 0.0893 0.9166 sion of algorithm e	bubbleSort 0.0051 0.184 13.2808	0.0019 0.0067 0.0939	lysis will focu	0 200	input		800 1000	Time taken			1000	
10 100 1000 Section 3 : Theoretical discuss Discussion : We can make a few o When input si	shellSort 0.0036 0.0893 0.9166 sion of algorithm e	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Sh	0.0019 0.0067 0.0939	lysis will focus	o 200	input		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell	shellSort 0.0036 0.0893 0.9166 sion of algorithm elbservations fro ize increases e 1). However, oil	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtinormously, Shriginal and modexes that are	0.0019 0.0067 0.0939 tained. Our analell Sort does no dified bubble so far apart, while	lysis will focust increase its rt 2 increases Bubble sort s	s on unsorted a time largely to s largely. This is waps 2 items a	input		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, oil Sort swaps included the store of th	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Shriginal and modexes that are eature allowed	0.0019 0.0067 0.0939 tained. Our analell Sort does no dified bubble so far apart, while lelements in the	lysis will focus it increase its rt 2 increases Bubble sort s e array to get	s on unsorted a time largely to b largely. This is twaps 2 items a into its valid po	input input inrays. sort the source to the at once that issition		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, oi I Sort swaps integether. This fo	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Shriginal and mode deves that are leature allowed arisons. We ca	0.0019 0.0067 0.0939 tained. Our analell Sort does no dified bubble so far apart, while elements in the n verify this by o	lysis will focus to increase its rt 2 increases Bubble sort s e array to get checking the	s on unsorted a time largely to b largely. This is waps 2 items a into its valid podata from secti	input arrays. sort the s due to the at once that sition on 1. As		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t quicker than a the size of the	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, oi I Sort swaps increased together. This fo	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Shriginal and modexes that are leature allowed arisons. We calles, Shell sort's	0.0019 0.0067 0.0939 tained. Our analell Sort does no dified bubble so far apart, while lelements in the	lysis will focus to increase its rt 2 increases Bubble sort s e array to get checking the	s on unsorted a time largely to b largely. This is waps 2 items a into its valid podata from secti	input arrays. sort the s due to the at once that sition on 1. As		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t quicker than a the size of the largely compa	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, or 1 Sort swaps increased array increases erred to BubbleS	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Shriginal and modexes that are eature allowed arisons. We call es, Shell sort's Sort 1 and 2.	0.0019 0.0067 0.0939 tained. Our analell Sort does no diffied bubble so far apart, while lelements in the n verify this by a number of com	lysis will focus t increase its rt 2 increases Bubble sort s e array to get checking the parisons and	s on unsorted a time largely to s largely. This is waps 2 items a into its valid podata from secti swaps does no	input arrays. sort the s due to the arraysition on 1. As ot increase		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t quicker than a the size of the largely compa	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, or I Sort swaps increases e array increases ared to Bubbles compare graph	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Shriginal and modexes that are eature allowed arisons. We called the control of the	0.0019 0.0067 0.0939 tained. Our analell Sort does not diffied bubble of a rapart, while I elements in the n verify this by a number of com	lysis will focus t increase its rt 2 increases Bubble sort s e array to get checking the parisons and	s on unsorted a time largely to a largely. This is waps 2 items a into its valid podata from secti swaps does no with large value.	input arrays. sort the due to the at once that sistion on 1. As ot increase es of n.		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t quicker than a the size of the largely compa	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, or 1 Sort swaps increases e array increase array increase array increase compare graph , shellSort's tim	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Shriginal and modexes that are eature allowed arisons. We called the control of the	0.0019 0.0067 0.0939 tained. Our analell Sort does no diffied bubble so far apart, while lelements in the n verify this by a number of com	lysis will focus t increase its rt 2 increases Bubble sort s e array to get checking the parisons and	s on unsorted a time largely to a largely. This is waps 2 items a into its valid podata from secti swaps does no with large value.	input arrays. sort the due to the at once that sistion on 1. As ot increase es of n.		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t quicker than a the size of the largely compa Suppose we o From graph 1	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, oi I Sort swaps inc iogether. This fi adjacent compa- e array increase ared to Bubbles compare graph , shellSort's tim / O(n²).	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Shriginal and more dexes that are eature allowed arisons. We callow the complexity in	tained. Our analell Sort does no diffied bubble so far apart, while I elements in the n verify this by onumber of coming of a typical gross approximately	lysis will focust increase its rt 2 increases Bubble sort searray to get checking the parisons and bowth function of O(nlogn), who	s on unsorted a time largely to s largely. This is waps 2 items a into its valid podata from secti swaps does nowith large valunile BubbleSort	input arrays. sort the s due to the at once that sistion on 1. As ot increase es of n. 1 and 2 is		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t quicker than a the size of the largely compa Suppose we o From graph 1 approximately From graph 1	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, oi I Sort swaps inc iogether. This fi adjacent compa- e array increase ared to Bubbles compare graph , shellSort's tim / O(n²).	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtinormously, Shriginal and more dexes that are eature allowed arisons. We cales, Shell sort's Sort 1 and 2. 11 with a graphine complexity is that represer	0.0019 0.0067 0.0939 tained. Our analell Sort does no dified bubble so far apart, while lelements in the n verify this by onumber of comin of a typical gross approximately	lysis will focust increase its rt 2 increases Bubble sort searray to get checking the parisons and bowth function of O(nlogn), who	s on unsorted a time largely to s largely. This is waps 2 items a into its valid podata from secti swaps does nowith large valunile BubbleSort	input arrays. sort the s due to the at once that sistion on 1. As ot increase es of n. 1 and 2 is		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t quicker than a the size of the largely compa Suppose we From graph 1 approximately From graph 1 can conclude	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, oil Sort swaps increases array increase array increase ared to Bubbles compare graph , shellSort's tim y O(n²), , since the lines their time com	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtinormously, Shriginal and more dexes that are eature allowed arisons. We call es, Shell sort's Sort 1 and 2. 1 with a graphine complexity is that represer plexity is also services.	0.0019 0.0067 0.0939 tained. Our analell Sort does no dified bubble so far apart, while lelements in the n verify this by onumber of comin of a typical gross approximately	lysis will focust increase its rt 2 increases Bubble sort sea raray to get checking the parisons and with function of O(nlogn), who diffied bubble site increases are reconstructed in the control of the	s on unsorted a time largely to s largely. This is waps 2 items a into its valid podata from secti swaps does nowith large valunile BubbleSort	input arrays. sort the s due to the at once that sition on 1. As ot increase es of n. 1 and 2 is		800 1000	Time taken			1000	
Section 3 : Theoretical discuss Discussion : We can make a few o When input si array (Graph fact that Shell are adjacent t quicker than a the size of the largely compa Suppose we From graph 1 approximately From graph 1 can conclude Thus, the effice	shellSort 0.0036 0.0893 0.9166 sion of algorithm e bservations fro ize increases e 1). However, oil Sort swaps increases array increase array increase ared to Bubbles compare graph , shellSort's tim y O(n²), , since the lines their time com	bubbleSort 0.0051 0.184 13.2808 efficiency om the data obtenormously, Shriginal and more deves that are eature allowed arisons. We call eature allowed arisons. We call eature allowed arisons are complexity is sthat representative is that representative is also algorithms call	tained. Our analell Sort does no diffied bubble so far apart, while lelements in the n verify this by onumber of coming of a typical gross approximately at original and misimilar.	lysis will focust increase its rt 2 increases Bubble sort sea raray to get checking the parisons and with function of O(nlogn), who diffied bubble site increases are reconstructed in the control of the	s on unsorted a time largely to s largely. This is waps 2 items a into its valid podata from secti swaps does nowith large valunile BubbleSort	input arrays. sort the s due to the at once that sition on 1. As ot increase es of n. 1 and 2 is		800 1000	Time taken			1000	