· · ·	13	6	2	7		4	8	15	9	3	mergeSort( array , first, last ) if first >= last
· · · ·	· · · · ·			· · · · ·		· · · · ·	· · ·			· · · ·	return mid = (first+last)/2 mergeSort(array, first, mid)
· · ·	· · · · ·	· · · · ·	· · ·	· · · · ·	· · ·	· · · · ·	· · ·		· · · · ·	· · ·	//A mergeSort(array, mid+1, last) //B
· · ·	· · · · ·		· · ·	· · · · ·	· · ·	· · · · ·	· · ·	· · · ·	· · · · ·	· · ·	merge( array, first, mid, mid+1, last)
· · ·			· · ·	· · · · ·	· · ·	· · · · ·	· · ·		· · · · ·	· · · ·	//C
· · ·			• • •		· · ·	· · · · ·	· · ·		· · · · ·	• • •	
• • •	· · · · ·	· · · · ·	· · ·	· · · · ·	· · ·	· · · · ·	• • •	• • •	· · · · ·	· · ·	
	· · · · ·	· · · · ·	· · ·	· · · · ·	· · ·	· · · · ·	· · · ·		· · · · ·	· · ·	
· · ·	· · · · ·		· · ·	· · · · ·	· · ·	· · · · ·	· · ·			· · ·	
• • •	• • • •		• • •	· · · · ·	• • •	· · · · ·	• • •		· · · · ·	· · ·	

Wł (as co	/ 3 nat w ssum mple	ill ab e tha ted a	Z ove a t the ll it r	7 array call neede	look to me ed to	Like erges do)	at lir Sort	15 ne ma on th	9 arkec e lin	d A? e ab	ove	mero //A	geSort( array , first, last ) if first >= last return mid = (first+last)/2 mergeSort(array, first, mid mergeSort(array, mid+1, merge( array, first, mid, mid+1, last	d) last)
•••										<u> </u>		//C		
	• • • •		• • • •	• • •		• • •		• • •			• • •	,,, C		
• • •	• • • •		• • • •	• • •	• • • •	• • •	• • • •	• • •	• • • •		• • •	• • • • •		
0 0 0	0 0 0 0			0 0 0	0 0 0 0	0 0 0		0 0 0	• • • •	0 0 0	0 0 0	• • • • •		
													· · · · · · · · · · · · · · · · · · ·	
				0 0 0										
• • •	• • • •		• • • •	• • •	• • • •	• • •	• • • •			• • •				
• • •	• • • •		• • • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	• • •	• • •		· · · · · · · · · · · · · · · · · · ·	
• • •	• • • •		• • • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	• • •	• • •			
• • •	• • • •			0 0 0		0 0 0	• • • •	0 0 0	• • • •	• • •	0 0 0	0 0 0 0 0		
• • •			• • • •	0 0 0	• • • •	• • •	• • • •	0 0 0	• • • •		• • •			
• • •	• • • •			• • •	• • • •		• • • •	• • •	• • • •	• • •	• • •			
• • •	• • • •		• • • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	• • •	• • •			
• • •	• • • •		• • • •	0 0 0	• • • •	• • •		0 0 0	• • • •		• • •			
0 0 0	0 0 0 0	0 0 0		0 0 0	0 0 0 0	0 0 0	0 0 0 0	0 0 0	• • • •	0 0 0	0 0 0			
				0 0 0				0 0 0			• • •			
												0 0 0 0		

W (a	/ha iss om	/3 nt w um	vill	6 I al tha	oo at all	2 ve th it	e e ne	7 rra ca	ay II t		ok mo to	lik erç do	t ce ge: )	at	8 : lii	ne on	ஏ m th	arl	۹ lin	d A	3 \? ab		e		me //A		if m m	or firs id erg	t( re ges	arr >= (fii 50	ay Irn st rt(	, +l arı arı	fir: ast ay	st, t)/2 7, f	la 2 irs nic	st t, 1	) ni( 1,	d) las		
		22 24 10	· · ·		. e	7			• •		) . ) .	4	0		<b>?</b>	15	• •	19 19 1	•	~ ~ ~ ~ ~ ~		• •	· • •	· ·	· · · ·	•	m	erg	ge(	( a	rra	ау,	fii m	st nid	, m +1	nid L, I	, as†	t)	· ·	
· · · ·		· · ·	• •	• •	•	• •	• •		• • •	•	• •	• •	•	• • •	• •	•	• •	• •	•	• •	· · ·	• •	· • •	• • • •	· · ·	0			• • • • • •	• •	•	· ·	• •	•	•	• •	· ·	· ·	• •	
• • •	) 0 ) 0 ) 0	• •	• •	• •	•	• •	• •	•	• •	•	• •	• •	•	• •	• • • •	•	• •	• • • •	•	• •	• • • •	• •	• •	• •	• •	•		• • • •	• • • •	• •	•	• • • •	• •	•	•	• • • •	• •	• •	• •	
		• • • • • •	• • • • • •	• •	•	• •	• •	• •	• • • • • • • •	•	• • • • • •	• •	•	• • • • • •	• •	•	• • • • • •	• • • • • •	•	• • • • • •	• • • • • •	• •	• •	· ·	• •	•			• • • • • •	• •	•	• • • • • •	• •	•	•	• • • • • • • •	• • • • • •	• •	• •	
• • • •		• •	• •	• •	•	• •	• •	•	• •	•	• •	• •	•	• •	• •	•	• •	• •	•	• •	· ·	• •	•	• •	• •	•			• •	• •	•	• •	• •	•	•	· ·	· ·	• •	• •	
• • • •	0 0	• •	• •	• •	•	• •	• •	0	• • • • • • • •	0	• • •	• •	•	• • •	• • •	•	• • •	• • • • • • • • • • • • • • • • • • •	•	• • •	• • • •	• •	0	• •	• •	0		0 0 0 0 0 0	• •	• •	0	• •	• •	•	0	• •	• •	• •	• •	

13627	11 4 8 15 9	mergeSort( array , first, last ) う if first >= last return mid = (first+last)/2	
What will above array (assume that the call completed all it neede	look like at line marked to mergeSort on the line ed to do)	d A? mergeSort(array, first, mid) ne above //A mergeSort(array, mid+1, last) //B	
267111	348159	3 merge( array, first, mid, mid+1, last)	· · ·
· · · · · · · · · · · · · · · · · · ·		$\dots \dots $	
What will above array (assume that the call t completed all it neede	look like at line marked to mergeSort on the lin d to do)	d B? e above	
What will above array (assume that the call t completed all it neede	look like at line marked to mergeSort on the lin d to do)	d B? e above	
What will above array (assume that the call to completed all it neede	look like at line marked o mergeSort on the lin d to do)	d B? e above	

1.3 $6$ $2$ $7$ $11$ $4$ What will above array look like (assume that the call to merge completed all it needed to do) $2$ $6$ $7$ $11$ $13$ $4$ $2$ $6$ $7$ $11$ $13$ $4$	81593at line marked A?Sort on the line above81593	<pre>mergeSort( array , first, last )     if first &gt;= last         return     mid = (first+last)/2     mergeSort(array, first, mid) //A     mergeSort(array, mid+1, last) //B     merge( array, first, mid,         mid+1, last) //C</pre>
What will above array look like a	at line marked B?	
(assume that the call to merges completed all it needed to do)	Sort on the line above	
(assume that the call to merges completed all it needed to do) 2 6 7 11 13 3 1	Fort on the line above	1       1

Wh (as cor	/3 atw sum nple	<i>6</i> ill ab e tha ted a	2 ove it the ill it r	7 array call need	/ look to m ed to	4 k like lerge do)	at li Sort	15 ne m on t	9 harke he lir	3 d A? ne at	) ove	m 	erge if n A A 3	Sort( first nid = nerge	arra >= etur (firs Sort	y , f last n t+la (arra	irst, ist)/ ay, 1 ay, 1	las 2 īrst, mid	t ) , mi +1,	d) las	t.)
What (associated as a construction of the second se	2 at wi sume	ll ab e tha	7 ove a t the II it n	array call	3 look to m	/ like erge do)	2 at lii Sort	5 ne m on th	9 arke ne lir	3 d B? ne ab	ove		2	nerge	e(ar	ray,	first mic	, mi  +1,	d, las	t)	
Wh (as con	at wisumenplet	ll ab e tha ed a	7 ove a t the II it r	array call neede	3 look to m ed to	3 like erge do)	at lii	8 he li	9 arke ne al	d C?		<ul> <li></li></ul>									

Wh (as COI	13 natw sum nple	ill ab e tha ted a	2 oove at the all it i	7 array call need	look to m ed to	4 k like herge do)	at li Sort	15 ne m on tl	9 narke ne lir	3 d A? ne ab	ove	m ///A	erges if m m s m	Sort( first nid = nerge	arra >= etur (firs Sort	y , fi last n t+la (arra (arra	rst, l st)/2 iy, fii iy, m irst, mid-	ast ) rst, r nid+: mid, +1, la	nid) L, la ast)	st)	
Wh (as cor	at w sum nple	ill ab e tha ted a	ove t the ll it r	array call neede	look to m ed to	( like erge do)	at lii Sort	ne m on th	arke ne lin	d B? ie ab	ove		· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	<ul> <li></li></ul>	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·		2 2 0
· · · · ·	2 (		7		3	3	4	8	9	15	· · · · ·		· · · · ·	· · · ·	· · · ·	· · · · ·	· · ·	· · · ·	· · · ·		
Wh (as cor	at w sum nple	ill ab e tha ted a	ove t the II it r	array call neede	look to m ed to	( like erge do)	at lii on t	ne m he lii 7	arke ne al	d C? bove				· · · · · · · · · · · · · · · · · · ·		1.2	· · · · ·		· · · · · · · · · · · · · · · · · · ·	· · · ·	2
0 0 0 0 0 0	• • • •	• • • •	• • •	• • • •		• • • •	• • •	· · · · ·	· · · ·			· · / ·	. (),	•••				• • •	• • •	• • •	

WI wh	na ne	/ t w n t	yil tha	l a at	ç ab li re	DO' NE CL	2 ve is	a a iv	rra e c	iy ich		) ok d f	lil	, te		لاً t I v		l : e i y f	) ma irs	erk st l	۹ tin	d A				ner A	rg (	eS if f mi me	ort irs d = erg	( a re = (  e\$	arra >= tu (fir Sor	ay Ia rn st- t(a	, f st +la arr	firs ast ay ay	st, t)/2 r, f	la 2 irs nic	st t, i 1+	) mia 1,	d) las		
		• •		0	• • • • • • • •	•	• •	0 0 0	• •	• • •	· ·	0 4 4 0	· · ·	•	• • • • • •	•	· · ·	•		• • •		· · ·	· · ·	· · ·		TC		me	erg	e(	aı	ra	у, у,	fir m	rst nid	, n +1	nid L, I	, ast	E).	· · ·	•
		• •		•	• • • • • •	•	• •	•	• •	•	• • • • • •	•	• • • • • •	•	• • • • • •	0 0 0	• • • • • •	•				• •	• •	• •	•	• • • • • •	• •	• • •	• • • • • •	• •	•		• • • • • •	•	• • • • • •	• •	• •		• • • • • •	• • • • • •	•
		• •		0 0 0	• • • • • •	•	• •	•	• • •	•	• • • • • •	•	• • • • • •	•	• • • • • •	•	• • • • • •	•		•		• •		• • • • • •	•	• • • •	• •	0 0 0	• • • • • •	• •	•	• • • •	• • • • • •	•	• • • •	• •	0 0 0 0 0 0		• • • • • •	• • • • • •	•
		• •		•	· · ·	•	· · ·	•	• • •	•	· · ·	•	• • •	•	• • • • • •	•	• • • • • • • •	•		• •		· · ·	• •	· · ·	•	• • • • • • • •	• •	0 0 0 0	· · ·		•		· · ·	•	· · ·	• •	• • •		• • •	· · ·	•
				0 0 0	• • • • • •	•	· · ·	•	• • •	•	• • • • • •	•	• • • • • •	•	• • • • • • • •	0	• • • • • • • •	•				• •	• •	· · ·	•	• • • • • •	• •	0 0 0	• • • • • •		•		· · ·	•	• • • • • • • •	• • •	• •		• • • • • •	· · ·	•

W	ha ne	/ 3 n th este	ill ab nat lin ed re	2 ove a ne is cursiv	7 array reac ve ca	look hed f lls?	4 like or th	8 at lin e ve	15 ne ma ry firs	9 arked st tin	J A ne		mero //A //B	geSort( if first r mid = merge merge	array , >= last eturn (first+l Sort(arr	first, last ast)/2 ay, first, ay, mid+	) mid) -1, last)
		13	6	2	7		4	8	15	9	3	• • •	//C	merge	e( array,	first, mic mid+1,	last)
· · · ·	•	· · · · ·	<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · ·		.         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .	<ul> <li>.</li> <li>.</li></ul>	· · · · · ·					.         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .	.         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .	.         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .         .           .		
· · · ·	0 0 0 0	· · · ·	<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · ·		.         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .	<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · ·					.     .     .     .     .     .       .     .     .     .     .     .       .     .     .     .     .     .       .     .     .     .     .     .       .     .     .     .     .     .       .     .     .     .     .     .       .     .     .     .     .     .	.         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .	.         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .         .           .		.       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         . <t< td=""></t<>
	0					· · · · · · · · · · · · · · · · · · ·								·     ·     ·     ·       ·     ·     ·     ·       ·     ·     ·     ·       ·     ·     ·     ·       ·     ·     ·     ·       ·     ·     ·     ·       ·     ·     ·     ·       ·     ·     ·     ·       ·     ·     ·     ·	·         ·	·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·       ·         ·       ·       ·       ·       ·       ·       ·       ·       ·       ·       ·       ·       ·       ·       ·       ·       ·       · <td></td>	

· · · · ·	13	6	2	. 7		4	8	15	9	3	me	rgeSort( array , first, last ) if first >= last return
Wh whe	at wi en th	ll ab at lir	ove a ne is	array reacl	look ned f	like or th	at lir e vei	ne ma ry firs	arkec st tim	i A A ne	· · · · · · · · · · · · · · · · · · ·	mid = (first+last)/2 mergeSort(array, first, mid)
in r	neste	d rec	cursiv	/e ca	lls?	• • •	· · · · ·	 	· · · · ·	• • •	//B	mergeSort(array, mid+1, last)
· · · ·	13	6 6	2	7		4		15	9	3		merge( array, first, mid, mid+1, last)
Wha whe	at wil en tha	l abo at lin	ove a e is i	rray reach	look ned fo	like or the	at lin e ver	e ma y firs	rked t tim	B <sup>a</sup>		.       .
in n	este	d rec	ursiv	ve cal	ls?	• • •	· · · · ·	• • • •	· · · · ·	• • •	· · · · · · · · ·	
· · · · ·	· · · · ·	• • •	· · · ·	· · ·	• • • •	• • •	· · · · ·	· · ·	· · · · ·	· · ·	· · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · ·	· · · · ·	• • •		• • •	• • • • •	• • •	• • • • •	• • •	• • • • •	· · ·	· · · · · · · · ·	
• • • •	• • •	· · ·	· · · ·	· · ·	• • • • •	0 0 0 0 0 0	· · · · ·	· · ·	· · · ·	• • •	· · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
• • • •												

	13	6	2	7		4	8	15	9	3		nergeS if m	ort( a first : re id =	array >= la eturn (first-	, firs st ⊦last	st, la -)/2	ast)		· · · ·	
Wh whe in r	at wi en th neste	ll abo at lir d rec	ove a ne is cursiv	array reac /e ca	look hed f lls?	like or th	at lir e vei	ne ma ry firs	arkec t tim	l A ne	· · · · · //	/A M /B	erges erges	Sort(a Sort(a	irray	, fir , mi	st, r d+1	nid) L, la	st)	
· · · ·	13	6	2	7		4	8	15	9	3	· · · · · · ·	m /C	erge(	arra	y, fir m	st, r iid+	nid, 1, la	ast)	· · ·	•
1 A / Ia .																				
whe in n	en tha ested	l abc at lin d rec	ove a e is ursiv	rray reach re ca	look ned fo lls?	like a or the	at lin e ver	e ma y firs	rked t tim	B	.     .     .     .       .     .     .     .       .     .     .     .       .     .     .     .       .     .     .     .       .     .     .     .       .     .     .     .		· · · · · ·				· · · ·	· · · ·	· · ·	
whe in n	en tha ested 13	l abo at lin d rec 6	ove a e is ursiv	rray reach re ca 7	look ned fo lls?	like a or the 4	at lin e ver	e ma y firs 15	rked t tim 9	B Ie 3			.         .         .         .           .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .         .           .							
vvna whe in n	ested	l abo at lin d rec	ove a e is ursiv	irray reach re ca 7	look ned fo lls?	like a or the	at lin e ver	e ma y firs	rked t tim	B	.         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .		<ul> <li></li></ul>							

Wh whe in r	/ 3 at wi en th neste	ill ab nat lin ed red	2 ove a ne is cursiv	7 array reac ve ca	look hed IIs?	4 like for th	8 at lii e ve	l5 ne m ry fir	9 arke st tir	ح d A ne	//A	rgeSort( if first mid = merge merge	( array : >= las return = (first+ eSort(a eSort(a	, first, st last)/2 rray, f rray, r	last 2 irst, r nid+	) nid) 1, la	st)
Wha whe in n	/3 at wi en th este	6 II abo at lir d rec	Z ove a ne is cursiv	7 array reacl /e ca	 look ned f lls?	4 like or th	8 at lir e vei	15 ne ma ry firs	9 arkeo st tin	3 d B ne	//C	merge	e(array	, first mid	, mid +1, l	, ast)	<ul> <li></li> <li></li></ul>
· · · · ·	13	6	2	7		4	8	15	9	3		· · · · · · ·		· · · · ·	· · · · ·	• • •	· · ·
Wh wh in r	at w en th neste	ill ab nat li ed re	ove ne is cursi	array reac ve ca	/ lool hed alls?	< like for th	at li ne ve	ne m ery fir	arke st tir	d C ne					· · · · · ·		· · · ·
	· · · ·	· · · ·	· · · · ·	· · · ·	· · · · ·			· · · · · ·							· · · · ·	· · · ·	· · · ·

	13		2	7		4 4	8	15	9	3		merg	eSort( if first	array >= la return	, first st	, las	t.).	· ·	• •
Wh whe in r	at wi en th ieste	ll ab at lir d rec	ove a ne is cursiv	array reacl /e ca	look hed f lls?	like or th	at lir e ve	ne ma ry firs	arkeo st tim	A A ne		//A	mid = merge merge	eSort(a Sort(a	-last) rray, rray,	/2 first, mid·	, mio +1,	d) last	· · ·
· · · · ·	13	6	2			· · · · ·	8	15	9	3	·     ·     ·       ·     ·     ·       ·     ·     ·       ·     ·     ·       ·     ·     ·       ·     ·     ·	//B	merge	e( array	/, firs mi	t, mi d+1,	d, las	t)	· · · ·
Wha whe in n	at wil en tha estee	ll abo at lin d rec	ove a le is ursiv	irray reach re cal	look ned fo lls?	like or the	at lin e ver	ie ma y firs	arked t tim	IB ne	· · · · · ·			·     ·     ·     ·     ·       ·     ·     ·     ·     ·       ·     ·     ·     ·     ·       ·     ·     ·     ·     ·       ·     ·     ·     ·     ·       ·     ·     ·     ·     ·       ·     ·     ·     ·     ·       ·     ·     ·     ·     ·			· · · ·	· · ·	• • • • • •
· · · · ·	13	6	2	7		4	8	15	9	3	· · · ·	· · · · · ·				· · · · ·	• • •	• •	0 0 0 0
Wh wh in r	at wi en th neste	ill ab nat lin ed re	ove a ne is cursiv	array reac ve ca	look hed f Ills?	like for th	at lii ie ve	ne m ry fir	arkeo st tin	d C ne								· · ·	· · ·
	· · · ·	· · · ·	· · · · ·	· · · ·	· · · · · ·	· · · ·	· · · · ·	6	1.3	2	7		4	3 15	9	3		• • • • • •	• •

13 6 2 7 11 4 8 15 9 3 What will above array look like at line marked C when that line is reached for the very FIRST time in nested recursive calls?	mergeSort( array , first, last ) if first >= last return mid = (first+last)/2 mergeSort(array, first, mid) //A mergeSort(array, mid+1, last) //B
6 13 2 7 11 4 8 15 9 3 What will above array look like at line marked C when that line is reached for the very SECOND time in nested recursive calls?	merge( array, first, mid, mid+1, last) //C

• • •	13		2	7		4	8	15	9	3		mergeSort( array , first, last ) if first >= last return	•
Wł wł	nat w nen tl neste	vill al hat li ed re	oove ine is	array read	y loo ched alls?	k like for tł	at li ne ve	ne m ery Fl	narke RST	ed C time		mid = (first+last)/2 mergeSort(array, first, mid) //A	•
· · ·			2			· · · ·				· · · ·	• • • •	mergeSort(array, mid+1, last) //B	0
· · · ·						4			<u> </u>		· · · · ·	merge( array, first, mid, mid+1, last)	•
W w in	/hat w hen t nest	will a that ted r	ibove line i ecurs	e arra s rea sive o	ay loo acheo calls?	ok like d for t ?	e at the v	line r ery S	mark SECC	ed C ND t	ime	//C	
• • •	2	6	13	7		4	8	15	9	3	· · · · ·	<pre></pre>	•
	· · · · ·	• • •			· · ·	· · · · ·	• • •	· · · · ·	· · ·	· · ·	· · · · ·		•
• • • • • •	• • • •	0 0 0 0 0 0	• • • •	, , , , , , , , , , , , , , , , , , ,	• • •	• • • • •	o o o o o o	• • • • • • • • •	• • •	• • •	• • • •		•
	· · · · ·	· · · ·			· · · ·	· · · · ·	· · · ·	· · · · ·	· · · ·	· · · ·	· · · · ·	·       ·	•

1.362711481593What will above array look like at line marked C when that line is reached for the very FIRST time in nested recursive calls?	<pre>mergeSort( array , first, last )     if first &gt;= last         return     mid = (first+last)/2         mergeSort(array, first, mid) //A         mergeSort(array, mid+1, last)</pre>
6 13 2 7 11 4 8 15 9 3 What will above array look like at line marked C when that line is reached for the very SECOND time in nested recursive calls?	//B merge( array, first, mid, mid+1, last) //C
2 6 13 7 11 4 8 15 9 3 What will above array look like at line marked C when that line is reached for the very THIRD time in nested recursive calls?	
	.

136	2	7		4	8	15	9	3		mergeSort( array , first, last ) if first >= last
What will al when that l	oove ine is	array	y loo ched	k like for t	e at l he vo	ine n ery F	narke IRST	ed C time	<b>9</b>	mid = (first+last)/2 mergeSort(array, first, mid) //A
in nested re	ecursi	ve ca	alls?	• • • •		• • • •		0 0 0 0 0 0	· · · · · ·	mergeSort(array, mid+1, last)
6 13	2	7		4	8	15	9	3		//B merge( array, first, mid, mid+1 last)
What will a when that in nested r	ibove line i ecurs	e arra s rea sive c	ay loo icheo calls?	ok lik d for ?	ke at the N	line /ery \$	mark SECC	ced C OND	time	//C
26	13	7		4	8	15	9		· · · · · ·	· · · · · · · · · · · · · · · · · · ·
What will a when that in nested	above line recur	e arra is rea sive	ay lo ache calls	ok li d for ?	ke at the	line very	mar THIR	ked ( (D tir	C ne	.       .
26	13	7		4	8	15	9	3	· · · · · · ·	

• • • • • •	13	6		2	.7		4	8 8 8	15	9	3	mergeSort( array , first, last ) if first >= last
• • • • • •	Wha mer	at is geSo	the ort f	tota unct	l nun ion ii	nber o n the	of recu proce	ursive ss of s	calls sorting	made g the	e to array	return mid = (first+last)/2 mergeSort(array, first, mid)
• •	abov	ver	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	//A
• •	• • •	• • •	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	mergeSort(array, mid+1, last) //B
• •	• • •	• • •	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	merge( array, first, mid,
• •	• • •	• • •	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	mid+1, last)
• •	• • •	• • •	• •	• • •	• • •	• • •		• • •		• • •		//C
• •	• • •	• • •	o o	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	
• •	• • •	• • •	• •	0 0 0	• • •	0 0 0 0 0 0	• • • •	0 0 0	• • • •	• • •	• • • •	
• •	• • •	• • •	0 0	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	· · · · · · · · · · · · · · · · · · ·
• •	• • •	• • •	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	
• •	0 0 0	• • •	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	
• •	• • •	• • •	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	
• •	0 0 0	• • •	0 0	0 0 0	0 0 0	0 0 0	· · · · ·	• • •	· · · · ·	0 0 0	0 0 0 0	
• •	• • •	• • •	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	· · · · · · · · · · · · · · · · · · ·
• •	• • •	• • •	• •	• • •	• • •	• • •	• • • •	• • •	• • • •	• • •	• • • •	· · · · · · · · · · · · · · · · · · ·
• •	0 0 0	0 0 0	• •	0 0 0	0 0 0	0 0 0		0 0 0	0 0 0 0	0 0 0	0 0 0 0	

	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	•	• •	• •		• •	• •			•	• •	• •	• •	• • •	• •	•	• •	• •	• •	•	• •	• •	• •	• •	•
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•	• • •	٠	• •	• •	• •	• • •	• •	•	• •	• •	• •	•	• •	• •	• •	• •	•
•	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •		• •	• •	•	• • •	•	• • •	• •	• •	• • •	• •		• •	• •	• •	•	• •	• •	• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	•	0 0	• •		• •	• •				• •	• •	• •	• • •	• •	•	• •	• •			• •	• •	0 0	• •	-
	• •	• •	• •	• •		• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •		• •	• •			•	• •	• •	• •	• • •	• •	•	• •	• •			• •		• •	• •	-
	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	0	• •	• •	•	• • •	•	• •	• •	• •	• • •	• •	0	• •	• •	• •		• •	• •	• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•	• • •	•	• • •	• •	• •	• • •	• •	•	• •	• •	• •		• •	• •	• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•		• •	• •	0	• •	• •			•	• • •	• •	• •	• • •	• •	0	• •	• •	• •		• •	• •	• •	• •	
	• •	• •	• •		• •	• •	• •			• •	• •	• •	• •	•		• •	• •		• •	• •			•	• • •	• •	• •		• •		• •	• •			• •		• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •		• •	• •	• •	• •	•		• •	• •		• •	• •		• • •	•	• • •	• •	• •	• • •	• •	•	• •	• •	• •		• •		• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	•	• •		•	• •	• •		• •	٠	• • •	• •	• •		• •	•	• •	• •			• •		• •	• •	
•	• •	• •	• •	• •	• •	• •	• •	• • •		• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•	• •	•	• • •	• •	• •		• •	•	• •	• •			• •		• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •		• •	• •	• •	• •	•	•	• •	• •	۰	• •	• •		• • •	٠	• • •	• •	• •	• • •	• •	•	• •	• •	• •		• •		• •	• •	
	• •		0 0	• •		• •	• •				• •	• •	• •			0 0	• •		• •	• •			0	• • •				• •		• •	• •		•	• •		0 0	• •	-
	• •		• •		• •	• •		• • •		• •	• •	• •	• •	•		• •			• •	• •		• • •	٠	• • •	• •	• •		• •		• •	• •			• •		• •	• •	-
	• •		• •			• •	• •	• • •		• •	• •	• •	• •	•	•	• •			• •		•		•	• • •		• •		• •	•	• •	• •			• •		• •	• •	
	• •		• •			• •	• •	• • •		• •	• •	• •	• •	•		• •			• •		•		•	• • •	• •	• •		• •		• •	• •			• •		• •	• •	
	• •	• •	• •		• •	• •		• • •	• •		• •	• •	• •			• •			• •	• •			•			• •		• •		• •				• •		• •	• •	
	• •	• •	• •	• •	• •	• •	• •			• •	• •	• •	• •	•		• •	• •		• •	• •			•	• • •	• •	• •		• •		• •	• •		•	• •		• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•		• •	• •		• •	• •		• • •	•	• • •	• •	• •	• • •	• •	•	• •	• •	• •		• •		• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•	• • •	•	• • •	• •	• •	• • •	• •	•	• •	• •	• •		• •	• •	• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•		• •	• •	0	• •	• •			•	• • •	• •	• •	• • •	• •	0	• •	• •	• •		• •	• •	• •	• •	
	• •	• •	• •	• •	• •	• •	0 0	• • •	• •	0 0	• •	• •	0 0	•		• •	• •	0	• •	• •				• • •	• •	0 0	• • •	• •	0	• •	• •	• •		• •		• •	0 0	
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	0 0	• •	• •	• •	•		• •	• •	0	• •	• •			•	• •	• •	0 0	• • •	• •	0	• •	• •	• •		• •	• •	• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	0	• •	• •	•		•	• •	• •	• •	• • •	• •	0	• •	• •	• •		• •	• •	• •	• •	-
•	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•			• •	• •	• •	• • •	• •	0	• •	• •	• •		• •	• •	• •	• •	
0	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •		• •	• •			•	• •	• •	• •	• • •	• •	0	• •	• •	• •		• •	• •	• •	• •	-
	• •	• •	0 0	• •	• •	• •	• •	• • •		• •	• •	0 0	• •		• •	0 0	• •		0 0	• •			0			• •		• •		• •	• •	• •	•	• •	• •	0 0	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •		• •	• •			•	• •	• •	• •	• • •	• •	•	• •	• •			• •	• •	• •	• •	-
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•	• • •	٠	• •	• •	• •	• • •	• •		• •	• •	• •	•	• •	• •	• •	• •	
•	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	٠	• •	• •	•	• • •	٠	• •	• •	• •	• • •	• •	٠	• •	• •	• •	•	• •	• •	• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	• •	• •	• •		• •	• •			•	• • •	• •	• •	• • •	• •		• •	• •	• •	•	• •	• •	• •	• •	-
	0 0	• •	0 0	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	• •	0 0	• •		• •	• •		• • •	0	• •	• •	• •	• • •	• •		• •	• •	• •	•	• •	• •	0 0	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	• •	• •	• •	0	• •	• •		• • •	٠	• •	• •	• •	• • •	• •	•	• •	• •	• •	•	• •	• •	• •	• •	-
	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•	• •	٠	• •	• •	• •	• • •	• •	٠	• •	• •	• •	•	• •	• •	• •	• •	
•	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	• •	• •	• •		• •	• •	•	• • •	۰	• • •	• •	• •	• • •	• •	0	• •	• •	• •	•	• •	• •	• •	• •	•
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	• •	• •	• •	• •	•	• •	• •	• •		• •	• •			0	• • •	• •	0 0	• • •	• •	0	• •	• •	• •	•	• •	• •	• •	• •	
	• •	• •	• •	• •	• •	• •	• •	• • •	• •	0 0	• •	• •	0 0	•		• •	• •	0	• •	• •			•	• •	• •	0 0	• • •	• •	0	• •	• •	• •	•	• •	• •	0 0	• •	•
•	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	0	• •	• •	•	• • •	•	• •	• •	• •	• • •	• •	0	• •	• •	• •	•	• •	• •	• •	• •	
•	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•	• • •	•	• •	• •	• •	• • •	• •	•	• •	• •	• •	•	• •	• •	• •	• •	•
•	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •	•	• •	• •	•	• •		• •	• •	• •	• • •	• •	•	• •	• •	• •		• •	• •	• •	• •	•
•	• •	• •	• •	• •	• •	• •	• •	• • •	•	• •	• •	• •	• •	•	•	• •	• •		• •	• •		• • •	•	• •	• •	• •	• • •	• •	•	• •	• •	• •		• •	• •	• •	• •	

mS(0,9)mS(5, ?)mS(0, 4)MS(5,7) MS(8,7)mS(0,2) mS(3,4)mS(0,1)mS(2,2)mS(3,3)mS(4,4)mS(5,6)mS(7,7)mS(8,8)mS(9,7)mS(0,0) mS(1,1)m5(5,5)mS(6,6)

13 $6$ $2$ $7$ What is the total number $I$ $g$ What is the total number $I$ $g$	11 4 8 15 9 3 mber of recursive calls made to in the process of sorting the arra mber of calls made to merge ess of sorting the array above?	<pre>mergeSort( array , first, last )     if first &gt;= last         return     mid = (first+last)/2     mergeSort(array, first, mid) //A     mergeSort(array, mid+1, last) //B     merge( array, first, mid,         mid+1, last) //C</pre>

13 6 2 7 11 4 8 15 9 3 What is the total number of recursive calls made to mergeSort function in the process of sorting the array above? 19	<pre>mergeSort( array , first, last )     if first &gt;= last         return     mid = (first+last)/2         mergeSort(array, first, mid) //A         mergeSort(array, mid+1, last) //B</pre>
What is the total number of calls made to merge function in the process of sorting the array above? $g_{just count}$	merge( array, first, mid, mid+1, last) //C
in the recursion	free)
	.
	1       1

13	6	2	7		4	8 15	9	3	] r	mergeSort( array , first, last ) if first >= last
Wha merg abov	t is th geSort 'e?	e tota funct	l num	nber of n the pr	recurs	sive calls of sortin	made g the	e to array		return mid = (first+last)/2 mergeSort(array, first, mid)
· · · · · ·	· · · · ·		<u>}</u>	· · · ·	· · · · ·		· · · · ·	• • •		//A mergeSort(array, mid+1, last) //B
Wha func	t is th tion ir	e tota 1 the	al nun oroce	nber of ss of so	calls orting	made to the array	merg abov	e e?		merge( array, first, mid, mid+1, last) //C
	· · · · ·	9.	( -	just ped	t ce	ount ooir	of	· · · · · · · · · · · · · · · · · · ·		
· · · · · ·	· · · · ·	· · · ·	· · · · · ·	n t	he	rea	178	, <b>,</b> , , , , , , , , , , , , , , , , ,	n fre	ee)
Wha give	t is th n time	e larg e in th	jest n ne pro	umber cess of	of fra f sortii	mes for t ng the ar	he me ray ab	ergeS ove?	ort functi	tion on a stack at any
· · · · · · ·	· · · · · ·	· · · ·	· · · · · ·	.			· · · · · ·	· · · ·	.	

rst, mid)
nid+1, last)
mid, +1, last)
·       ·
.       .
?e)