

use key-indexed counting on first character

recursively sort subarrays

	count frequencies			transform counts to indices			distribute and copy back			indices at completion of distribute phase					
0	she	0	0	0	0	0	are	0	0	0	sort(a, 0, 0, 1);	0	are		
1	sells	1	a	0	1	a	0	1	a	1	sort(a, 1, 1, 1);	1	by		
2	seashells	2	b	1	2	b	1	2	she	2	b	2	sort(a, 2, 1, 1);	2	sea
3	by	3	c	1	3	c	2	3	sells	3	c	2	sort(a, 2, 1, 1);	3	seashells
4	the	4	d	0	4	d	2	4	seashells	4	d	2	sort(a, 2, 1, 1);	4	seashells
5	sea	5	e	0	5	e	2	5	sea	5	e	2	sort(a, 2, 1, 1);	5	sells
6	shore	6	f	0	6	f	2	6	shore	6	f	2	sort(a, 2, 1, 1);	6	sells
7	the	7	g	0	7	g	2	7	shells	7	g	2	sort(a, 2, 1, 1);	7	she
8	shells	8	h	0	8	h	2	8	she	8	h	2	sort(a, 2, 1, 1);	8	she
9	she	9	i	0	9	i	2	9	sells	9	i	2	sort(a, 2, 1, 1);	9	shells
10	sells	10	j	0	10	j	2	10	surely	10	j	2	sort(a, 2, 1, 1);	10	shore
11	are	11	k	0	11	k	2	11	seashells	11	k	2	sort(a, 2, 1, 1);	11	surely
12	surely	12	l	0	12	l	2	12	the	12	l	2	sort(a, 2, 1, 1);	12	the
13	seashells	13	m	0	13	m	2	13	the	13	m	2	sort(a, 2, 1, 1);	13	the
		14	n	0	14	n	2			14	n	2	sort(a, 2, 1, 1);		
		15	o	0	15	o	2			15	o	2	sort(a, 2, 1, 1);		
		16	p	0	16	p	2			16	p	2	sort(a, 2, 1, 1);		
		17	q	0	17	q	2			17	q	2	sort(a, 2, 1, 1);		
		18	r	0	18	r	2			18	r	2	sort(a, 2, 11, 1);		
		19	s	0	19	s	2			19	s	12	sort(a, 12, 13, 1);		
		20	t	10	20	t	12			20	t	14	sort(a, 14, 13, 1);		
		21	u	2	21	u	14			21	u	14	sort(a, 14, 13, 1);		
		22	v	0	22	v	14			22	v	14	sort(a, 14, 13, 1);		
		23	w	0	23	w	14			23	w	14	sort(a, 14, 13, 1);		
		24	x	0	24	x	14			24	x	14	sort(a, 14, 13, 1);		
		25	y	0	25	y	14			25	y	14	sort(a, 14, 13, 1);		
		26	z	0	26	z	14			26	z	14	sort(a, 14, 13, 1);		
		27		0	27		14			27	1		sort(a, 14, 13, 1);		

start of s subarray

1 + end of s subarray