

Csci 41: Introduction to Data Structures

Lab Exercise 6

July 15, 2014

Objectives:

- Merge Sort and Quick Sort
 - **Use the description of the algorithms to convert them into source code. Please try it without referencing to any material first. Only refer to material when really need it. This lab is to exercise how to convert “ideas/algorithms” into source code.**

Exercise Summary:

- Implement Merge Sort and Quick sort WITHOUT referencing ANY materials except the animation PDF attached.

When the lab session is over today, compress your **cpp** file(s) (and **header** file(s), if any) into a **single zip file** called YourLastName-Lab6.zip (e.g., Liu-Lab6.zip) and upload it to Blackboard. (**DO NOT** upload the entire project). Note that if you do not submit whatever you have done by the end of lab, you will get 0.5% penalty. You may continue to work on your lab after submission, but you must turn in whatever you have by the end of lab session.

-

Merge Sort

```

    sort(a, 0, 15)
    sort left half sort(a, 0, 7)
        sort(a, 0, 3)
            sort(a, 0, 1)
                merge(a, 0, 0, 1)
            sort(a, 2, 3)
                merge(a, 2, 2, 3)
            merge(a, 0, 1, 3)
        sort(a, 4, 7)
            sort(a, 4, 5)
                merge(a, 4, 4, 5)
            sort(a, 6, 7)
                merge(a, 6, 6, 7)
            merge(a, 4, 5, 7)
        merge(a, 0, 3, 7)
    sort right half sort(a, 8, 15)
        sort(a, 8, 11)
            sort(a, 8, 9)
                merge(a, 8, 8, 9)
            sort(a, 10, 11)
                merge(a, 10, 10, 11)
            merge(a, 8, 9, 11)
        sort(a, 12, 15)
            sort(a, 12, 13)
                merge(a, 12, 12, 13)
            sort(a, 14, 15)
                merge(a, 14, 14, 15)
            merge(a, 12, 13, 15)
        merge(a, 8, 11, 15)
    merge results merge(a, 0, 7, 15)

```

	lo	hi	a[]															
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			M	E	R	G	E	S	O	R	T	E	X	A	M	P	L	E
merge(a,	0,	0, 1)	E	M	R	G	E	S	O	R	T	E	X	A	M	P	L	E
merge(a,	2,	2, 3)	E	M	G	R	E	S	O	R	T	E	X	A	M	P	L	E
merge(a,	0,	1, 3)	E	G	M	R	E	S	O	R	T	E	X	A	M	P	L	E
merge(a,	4,	4, 5)	E	G	M	R	E	S	O	R	T	E	X	A	M	P	L	E
merge(a,	6,	6, 7)	E	G	M	R	E	S	O	R	T	E	X	A	M	P	L	E
merge(a,	4,	5, 7)	E	G	M	R	E	O	R	S	T	E	X	A	M	P	L	E
merge(a,	0,	3, 7)	E	E	G	M	O	R	R	S	T	E	X	A	M	P	L	E
merge(a,	8,	8, 9)	E	E	G	M	O	R	R	S	E	T	X	A	M	P	L	E
merge(a,	10,	10, 11)	E	E	G	M	O	R	R	S	E	T	A	X	M	P	L	E
merge(a,	8,	9, 11)	E	E	G	M	O	R	R	S	A	E	T	X	M	P	L	E
merge(a,	12,	12, 13)	E	E	G	M	O	R	R	S	A	E	T	X	M	P	L	E
merge(a,	14,	14, 15)	E	E	G	M	O	R	R	S	A	E	T	X	M	P	E	L
merge(a,	12,	13, 15)	E	E	G	M	O	R	R	S	A	E	T	X	E	L	M	P
merge(a,	8,	11, 15)	E	E	G	M	O	R	R	S	A	E	E	L	M	P	T	X
merge(a,	0,	7, 15)	A	E	E	E	E	G	L	M	M	O	P	R	R	S	T	X

Trace of merge results for top-down mergesort

	lo	j	hi	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
initial values				Q	U	I	C	K	S	O	R	T	E	X	A	M	P	L	E
random shuffle				K	R	A	T	E	L	E	P	U	I	M	Q	C	X	O	S
	0	5	15	E	C	A	I	E	K	L	P	U	T	M	Q	R	X	O	S
	0	3	4	E	C	A	E	I	K	L	P	U	T	M	Q	R	X	O	S
	0	2	2	A	C	E	E	I	K	L	P	U	T	M	Q	R	X	O	S
	0	0	1	A	C	E	E	I	K	L	P	U	T	M	Q	R	X	O	S
	1		1	A	C	E	E	I	K	L	P	U	T	M	Q	R	X	O	S
	4		4	A	C	E	E	I	K	L	P	U	T	M	Q	R	X	O	S
	6	6	15	A	C	E	E	I	K	L	P	U	T	M	Q	R	X	O	S
	7	9	15	A	C	E	E	I	K	L	M	O	P	T	Q	R	X	U	S
	7	7	8	A	C	E	E	I	K	L	M	O	P	T	Q	R	X	U	S
	8		8	A	C	E	E	I	K	L	M	O	P	T	Q	R	X	U	S
	10	13	15	A	C	E	E	I	K	L	M	O	P	S	Q	R	T	U	X
	10	12	12	A	C	E	E	I	K	L	M	O	P	R	Q	S	T	U	X
	10	11	11	A	C	E	E	I	K	L	M	O	P	Q	R	S	T	U	X
	10		10	A	C	E	E	I	K	L	M	O	P	Q	R	S	T	U	X
	14	14	15	A	C	E	E	I	K	L	M	O	P	Q	R	S	T	U	X
	15		15	A	C	E	E	I	K	L	M	O	P	Q	R	S	T	U	X
result				A	C	E	E	I	K	L	M	O	P	Q	R	S	T	U	X

Quicksort trace (array contents after each partition)