

CSCI 303: Algorithms, HW 7

Due: 3:30 pm, Wednesday, 10/24

1. Problem 1

Original	42	57	7	40	83	78	86	89	80	91	79	84	After Line:	Swap/Shift Count:
Quicksort(0, 11)	42	57	7	40	83	78	86	89	80	91	79	84	9	
Quicksort(0, 11)	42	57	7	40	83	79	86	89	80	91	78	84	13	1
Quicksort(0, 11)	42	57	7	40	78	79	86	89	80	91	83	84	27	1
Quicksort(0, 3)	42	57	7	40	78	79	86	89	80	91	83	84	9	
InsertionSort p = 1	42	57	7	40	78	79	86	89	80	91	83	84		0
InsertionSort p = 2	7	42	57	40	78	79	86	89	80	91	83	84		2
InsertionSort p = 3	7	40	42	57	78	79	86	89	80	91	83	84		2
Quicksort(5, 11)	7	40	42	57	78	79	86	89	80	91	83	84	9	
Quicksort(5, 11)	7	40	42	57	78	79	86	89	83	91	80	84	13	1
Quicksort(5, 11)	7	40	42	57	78	79	80	89	83	91	86	84	27	1
Quicksort(5, 5)	7	40	42	57	78	79	80	89	83	91	86	84	9	
Quicksort(7, 11)	7	40	42	57	78	79	80	89	83	91	86	84	9	
InsertionSort p = 8	7	40	42	57	78	79	80	83	89	91	86	84		1
InsertionSort p = 9	7	40	42	57	78	79	80	83	89	91	86	84		0
InsertionSort p = 10	7	40	42	57	78	79	80	83	86	89	91	84		2
InsertionSort p = 11	7	40	42	57	78	79	80	83	84	86	89	91		3

The number of swaps required by quick sort was 10. The number of shifts required by insertion sort was 4.

2. Problem 2

Original	42	57	7	40	83	78	86	89	80	91	79	84	After Line:	Swa/Shiftp Count:
Quicksort(0, 11)	42	57	7	40	83	78	86	89	80	91	79	84	9	
Quicksort(0, 11)	42	57	7	40	83	79	86	89	80	91	78	84	13	1
Quicksort(0, 11)	42	57	7	40	78	79	86	89	80	91	83	84	27	1
Quicksort(0, 3)	42	57	7	40	78	79	86	89	80	91	83	84	9	
Quicksort(0, 3)	40	7	42	57	78	79	86	89	80	91	83	84	13	3
Quicksort(0, 3)	40	7	42	57	78	79	86	89	80	91	83	84	27	1
Quicksort(0, 1)	40	7	42	57	78	79	86	89	80	91	83	84	9	
InsertionSort p = 1	7	40	42	57	78	79	86	89	80	91	83	84		1
Quicksort(3, 3)	7	40	42	57	78	79	86	89	80	91	83	84	9	
Quicksort(5, 11)	7	40	42	57	78	79	86	89	80	91	83	84	9	
Quicksort(5, 11)	7	40	42	57	78	79	86	89	83	91	80	84	13	1
Quicksort(5, 11)	7	40	42	57	78	79	80	89	83	91	86	84	27	1
Quicksort(5, 5)	7	40	42	57	78	79	80	89	83	91	86	84	9	
Quicksort(7, 11)	7	40	42	57	78	79	80	89	83	91	86	84	9	
Quicksort(7, 11)	7	40	42	57	78	79	80	84	83	86	89	91	13	3
Quicksort(7, 11)	7	40	42	57	78	79	80	84	83	86	89	91	27	1
Quicksort(7, 9)	7	40	42	57	78	79	80	84	83	86	89	91	9	
Quicksort(7, 9)	7	40	42	57	78	79	80	83	84	86	89	91	13	2
Quicksort(7, 9)	7	40	42	57	78	79	80	83	84	86	89	91	27	1
Quicksort(7, 7)	7	40	42	57	78	79	80	83	84	86	89	91	9	
Quicksort(9, 9)	7	40	42	57	78	79	80	83	84	86	89	91	9	
Quicksort(11, 11)	7	40	42	57	78	79	80	83	84	86	89	91	9	

The number of swaps required by quick sort was 15. The number of shifts required by insertion sort was 1.

3. Problem 3 A strategy that relies mostly on quick sort results in a high number of swaps, and a strategy that relies only on insertion sort results in a high number, n = number elements, of shifts. By using a mixed strategy it minimizes both the number of swaps and shifts done. The actual data movement required to effect the shift is greater than the swap. For the shift it requires at least $n + 2$ assignments, whereas swap could occur at least twice and at most $n + 5$ times.
4. Problem 4

Original	42	57	7	40	83	78	86	89	80	91	79	84	After Line:	Swap Count:
Quicksort(0, 11)	42	57	7	40	83	78	86	89	80	91	79	84	9	
Quicksort(0, 11)	42	57	7	40	83	78	86	89	80	91	79	84	13	
Quicksort(0, 11)	42	40	7	79	83	78	86	89	80	91	57	84	27	2
Quicksort(0, 2)	42	40	7	79	83	78	86	89	80	91	57	84	9	
InsertionSort p = 1	40	42	7	79	83	78	86	89	80	91	57	84		1
InsertionSort p = 2	7	40	42	79	83	78	86	89	80	91	57	84		2
Quicksort(4, 11)	7	40	42	79	83	78	86	89	80	91	57	84	9	
InsertionSort p = 5	7	40	42	78	79	83	86	89	80	91	57	84		2
InsertionSort p = 6	7	40	42	78	79	83	86	89	80	91	57	84		0
InsertionSort p = 7	7	40	42	78	79	83	86	89	80	91	57	84		0
InsertionSort p = 8	7	40	42	78	79	80	83	86	89	91	57	84		3
InsertionSort p = 9	7	40	42	78	79	80	83	86	89	91	57	84		0
InsertionSort p = 10	7	40	42	57	78	79	80	83	86	89	91	84		7
InsertionSort p = 11	7	40	42	57	78	79	80	83	84	86	89	91		3

5. Problem 5

Original	42	57	7	40	83	78	86	89	80	91	79	84	After Line:	Swap Count:
Quicksort(0, 11)	42	57	7	40	83	78	86	89	80	91	79	84	9	
Quicksort(0, 11)	42	57	7	40	83	78	86	89	80	91	79	84	13	
Quicksort(0, 11)	42	57	7	40	83	78	80	79	86	91	89	84	27	2
Quicksort(0, 6)	42	57	7	40	83	78	80	79	86	91	89	84	9	
InsertionSort p = 1	42	57	7	40	83	78	80	79	86	91	89	84		0
InsertionSort p = 2	7	42	57	40	83	78	80	79	86	91	89	84		2
InsertionSort p = 3	7	40	42	57	83	78	80	79	86	91	89	84		2
InsertionSort p = 4	7	40	42	57	83	78	80	79	86	91	89	84		0
InsertionSort p = 5	7	40	42	57	78	83	80	79	86	91	89	84		1
InsertionSort p = 6	7	40	42	57	78	80	83	79	86	91	89	84		1
Quicksort(8, 11)	7	40	42	57	78	80	83	79	86	91	89	84	9	
InsertionSort p = 9	7	40	42	57	78	80	83	79	86	91	89	84		0
InsertionSort p = 10	7	40	42	57	78	80	83	79	86	89	91	84		1
InsertionSort p = 11	7	40	42	57	78	80	83	79	84	86	89	91		3

6. Problem 6

Original	3	4	1	2	5	After Line:	Swap Count:
Quicksort(0, 4)	3	4	1	2	5	9	
InsertionSort p = 1	3	4	1	2	5	0	0
InsertionSort p = 2	1	3	4	2	5	0	2
InsertionSort p = 3	1	2	3	4	5	0	2
InsertionSort p = 4	1	2	3	4	5	0	0