Wong Ka Ngai UID:3035568881

Q2.3 (1)
Bubble sort: 3185us
recursiveMergesort: 462us
speedup=3185/462=6.8939

```
knwong@workbench:~/as1/q3$ ./main 1 256
This is the BEGINNING of the program.
n: 1; max_num: 256.
Sort (((4^n)*max_num) = 1024 integers.
Start timing recursiveMergesort...
End timing.
The elapsed time (us) is 462
Start timing bubble_sort...
End timing.
The elapsed time (us) is 3185
The sort result by merge sort is corrent, verified by bubble sort.
This is the END of the program.
knwong@workbench:~/as1/q3$
```

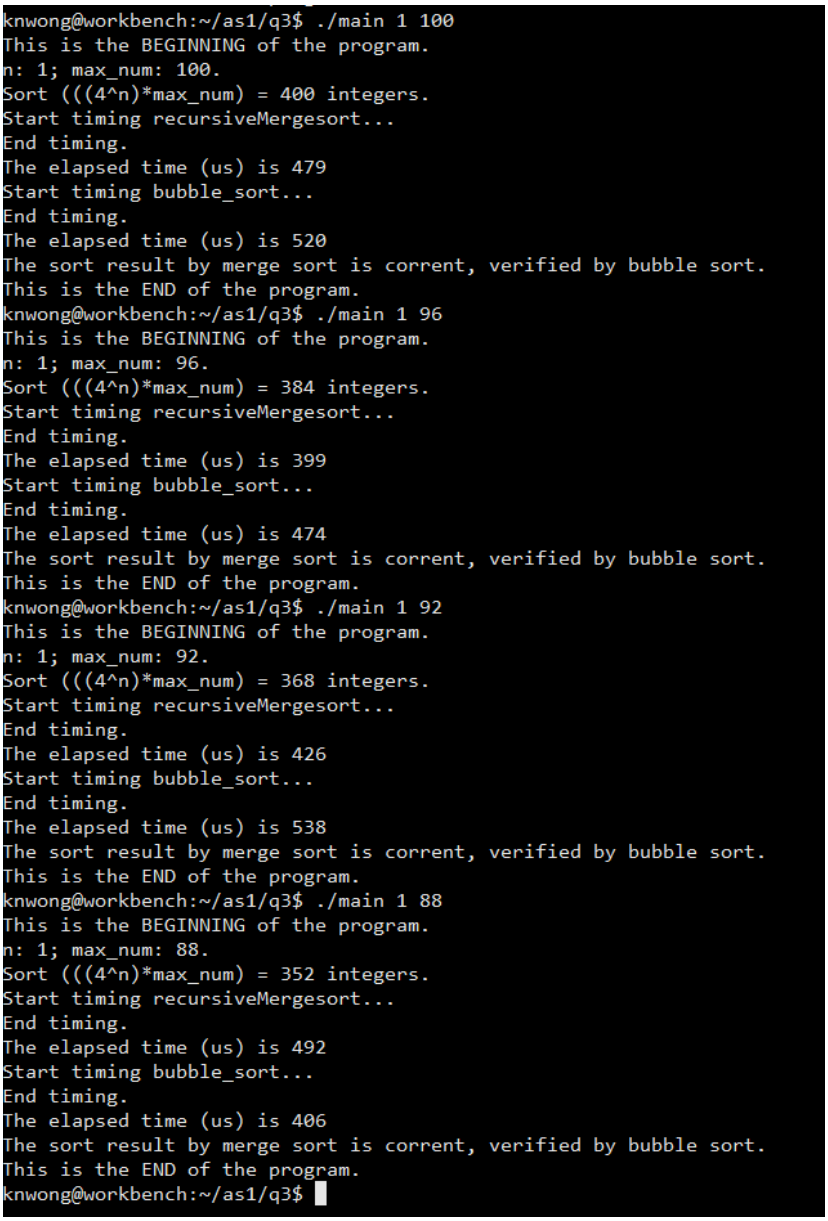
Q2.3 (2)
Bubble sort: 17207350us
recursiveMergesort: 3705us
speedup=17207350/3705=4644.3589

```
knwong@workbench:~/as1/q3$ ./main 2 4096
This is the BEGINNING of the program.
n: 2; max_num: 4096.
Sort (((4^n)*max_num) = 65536 integers.
Start timing recursiveMergesort...
End timing.
The elapsed time (us) is 3705
Start timing bubble_sort...
End timing.
The elapsed time (us) is 17207350
The sort result by merge sort is corrent, verified by bubble sort.
This is the END of the program.
knwong@workbench:~/as1/q3$
```

Q2.3 (3)
Bubble sort: 17397014us
recursiveMergesort: 4072us
speedup=17397014/4072=4272.3511

```
knwong@workbench:~/as1/q3$ ./main 3 1024
This is the BEGINNING of the program.
n: 3; max_num: 1024.
Sort (((4^n)*max_num) = 65536 integers.
Start timing recursiveMergesort...
End timing.
The elapsed time (us) is 4072
Start timing bubble_sort...
End timing.
The elapsed time (us) is 17397014
The sort result by merge sort is corrent, verified by bubble sort.
This is the END of the program.
knwong@workbench:~/as1/q3$
```

Q2.3 (4)



(1) speedup=3185/462=6.8939
(2) speedup=17207350/3705=4644.3589
(3) speedup=17397014/4072=4272.3511

When n=1, max_num=92-100, 4-way merge-sort and sequential bubble sort runs in similar time but 4-way merge-sort is slightly faster. When max_num is less than 92, bubble sort is faster. When the number is getting bigger, 4-way merge-sort outperforms bubble sort, as seen in (1)(2)(3).

Let the total number of integers be n. The time complexity of bubble sort is $\Theta(n^2)$, but the time complexity of 4-way merge-sort is $\Theta(n\log_4 n)$. The ratio of time complexity (which also represents the speedup) is $\Theta(n/\log_4 n)$. So when n is small, the speedup is small and sequential bubble sort outperforms 4-way merge-sort. However, when n gets bigger, after some n, the speedup is huge and 4-way merge-sort is faster than sequential bubble sort.