

Timothy Zamora

Prof. Torosdagli

COP 3502C

March 23, 2024

<https://github.com/tgnzamora/Bonus-Program-Assignment>

Execution of the compareSortAlgorithms.c

```
Extra memory allocated, size: 40000
Extra memory allocated, size: 40000
-----
Dataset Size : 10000
-----
Selection Sort:
    runtime           : 0.2
    extra memory allocated : 0
    Data:
        1 44 123 359 456 586 636 676 690 883 1114 1125 1196 1247 1334 1338 1459 1525 1531 1549 1597 1599 1671 1680 1974 1980
        2112 2223 2524 2808 2809 2817 2960 2969 3046 3192 3394 3479 3640 3722 3870 4071 4542 4622 4737 4774 4903 5005 5144 5322 540
6 5493 5510 5521 5987 6009 6355 6557 6571 6819 6870 6885 6915 6919 6968 6988 7053 7054 7077 7086 7093 7173 7284 7367 7408 74
16 7675 7691 7945 8030 8166 8170 8754 9009 9180 9310 9513 9635 9636 9732 9753 9769 9816 9829 9836 10051 10171 10244 10448 10
464
        989664 989698 989761 989812 989856 989869 989883 989982 990148 990392 990465 990824 990870 990895 990913 991178 9912
17 991279 991470 991811 991821 991918 992088 992331 992460 992554 992630 992670 992795 992816 992816 992836 992902 993038 99
3173 993312 993513 993551 993561 993592 993607 993692 993866 993875 993888 993897 993921 994009 994013 994176 994189 994210
994486 994573 994606 994689 994694 994989 995172 995207 995265 995385 995457 995469 995513 995592 995963 996173 996186 99632
4 996378 996379 996508 996531 996537 996544 996690 996753 996842 996866 996994 997070 997290 997601 997785 997825 997873 998
165 998170 998240 998370 998835 999107 999415 999420 999606 999791 999810 999963 999990

Insertion Sort:
    runtime           : 0.1
    extra memory allocated : 0
    Data:
        1 44 123 359 456 586 636 676 690 883 1114 1125 1196 1247 1334 1338 1459 1525 1531 1549 1597 1599 1671 1680 1974 1980
        2112 2223 2524 2808 2809 2817 2960 2969 3046 3192 3394 3479 3640 3722 3870 4071 4542 4622 4737 4774 4903 5005 5144 5322 540
6 5493 5510 5521 5987 6009 6355 6557 6571 6819 6870 6885 6915 6919 6968 6988 7053 7054 7077 7086 7093 7173 7284 7367 7408 74
16 7675 7691 7945 8030 8166 8170 8754 9009 9180 9310 9513 9635 9636 9732 9753 9769 9816 9829 9836 10051 10171 10244 10448 10
464
        989664 989698 989761 989812 989856 989869 989883 989982 990148 990392 990465 990824 990870 990895 990913 991178 9912
17 991279 991470 991811 991821 991918 992088 992331 992460 992554 992630 992670 992795 992816 992816 992836 992902 993038 99
3173 993312 993513 993551 993561 993592 993607 993692 993866 993875 993888 993897 993921 994009 994013 994176 994189 994210
994486 994573 994606 994689 994694 994989 995172 995207 995265 995385 995457 995469 995513 995592 995963 996173 996186 99632
4 996378 996379 996508 996531 996537 996544 996690 996753 996842 996866 996994 997070 997290 997601 997785 997825 997873 998
165 998170 998240 998370 998835 999107 999415 999420 999606 999791 999810 999963 999990

Bubble Sort:
    runtime           : 0.4
    extra memory allocated : 0
    Data:
        1 44 123 359 456 586 636 676 690 883 1114 1125 1196 1247 1334 1338 1459 1525 1531 1549 1597 1599 1671 1680 1974 1980
        2112 2223 2524 2808 2809 2817 2960 2969 3046 3192 3394 3479 3640 3722 3870 4071 4542 4622 4737 4774 4903 5005 5144 5322 540
6 5493 5510 5521 5987 6009 6355 6557 6571 6819 6870 6885 6915 6919 6968 6988 7053 7054 7077 7086 7093 7173 7284 7367 7408 74
16 7675 7691 7945 8030 8166 8170 8754 9009 9180 9310 9513 9635 9636 9732 9753 9769 9816 9829 9836 10051 10171 10244 10448 10
464
        989664 989698 989761 989812 989856 989869 989883 989982 990148 990392 990465 990824 990870 990895 990913 991178 9912
17 991279 991470 991811 991821 991918 992088 992331 992460 992554 992630 992670 992795 992816 992816 992836 992902 993038 99
3173 993312 993513 993551 993561 993592 993607 993692 993866 993875 993888 993897 993921 994009 994013 994176 994189 994210
994486 994573 994606 994689 994694 994989 995172 995207 995265 995385 995457 995469 995513 995592 995963 996173 996186 99632
4 996378 996379 996508 996531 996537 996544 996690 996753 996842 996866 996994 997070 997290 997601 997785 997825 997873 998
165 998170 998240 998370 998835 999107 999415 999420 999606 999791 999810 999963 999990
```

```
Merge Sort:
    runtime           : 0.0
    extra memory allocated : 0
    Data:
      1 44 123 359 456 586 636 676 690 883 1114 1125 1196 1247 1334 1338 1459 1525 1531 1549 1597 1599 1671 1680 1974 1980
      2112 2223 2524 2808 2809 2817 2960 2969 3046 3192 3394 3479 3640 3722 3870 4071 4542 4622 4737 4774 4903 5005 5144 5322 540
6 5493 5510 5521 5987 6009 6355 6557 6571 6819 6870 6885 6915 6919 6968 6988 7053 7054 7077 7086 7093 7173 7284 7367 7408 74
16 7675 7691 7945 8030 8166 8170 8754 9009 9180 9310 9513 9635 9636 9732 9753 9769 9816 9829 9836 10051 10171 10244 10448 10
464
      989664 989698 989761 989812 989856 989869 989883 989982 990148 990392 990465 990824 990870 990895 990913 991178 9912
17 991279 991470 991811 991821 991918 992088 992331 992460 992554 992630 992670 992795 992816 992816 992836 992902 993038 99
3173 993312 993513 993551 993561 993592 993607 993692 993866 993875 993888 993897 993921 994009 994013 994176 994189 994210
994486 994573 994606 994689 994694 994989 995172 995207 995265 995385 995457 995469 995513 995592 995963 996173 996186 99632
4 996378 996379 996508 996531 996537 996544 996690 996753 996842 996866 996994 997070 997290 997601 997785 997825 997873 998
165 998170 998240 998370 998835 999107 999415 999420 999606 999791 999810 999963 999990

Heap Sort:
    runtime           : 0.0
    extra memory allocated : 0
    Data:
      1 44 123 359 456 586 636 676 690 883 1114 1125 1196 1247 1334 1338 1459 1525 1531 1549 1597 1599 1671 1680 1974 1980
      2112 2223 2524 2808 2809 2817 2960 2969 3046 3192 3394 3479 3640 3722 3870 4071 4542 4622 4737 4774 4903 5005 5144 5322 540
6 5493 5510 5521 5987 6009 6355 6557 6571 6819 6870 6885 6915 6919 6968 6988 7053 7054 7077 7086 7093 7173 7284 7367 7408 74
16 7675 7691 7945 8030 8166 8170 8754 9009 9180 9310 9513 9635 9636 9732 9753 9769 9816 9829 9836 10051 10171 10244 10448 10
464
      989664 989698 989761 989812 989856 989869 989883 989982 990148 990392 990465 990824 990870 990895 990913 991178 9912
17 991279 991470 991811 991821 991918 992088 992331 992460 992554 992630 992670 992795 992816 992816 992836 992902 993038 99
3173 993312 993513 993551 993561 993592 993607 993692 993866 993875 993888 993897 993921 994009 994013 994176 994189 994210
994486 994573 994606 994689 994694 994989 995172 995207 995265 995385 995457 995469 995513 995592 995963 996173 996186 99632
4 996378 996379 996508 996531 996537 996544 996690 996753 996842 996866 996994 997070 997290 997601 997785 997825 997873 998
165 998170 998240 998370 998835 999107 999415 999420 999606 999791 999810 999963 999990

Extra memory deallocated, size: 40000
Extra memory deallocated, size: 40000
Extra memory allocated, size: 400000
Extra memory allocated, size: 400000
-----
Dataset Size : 100000
-----
Selection Sort:
    runtime           : 19.7
    extra memory allocated : 0
    Data:
      5 6 64 122 171 369 401 499 667 794 1005 1281 1414 1822 2076 2342 2460 2545 2616 2620 2674 2752 2877 2932 3070 3341 3
353 3481 3561 3589 3649 3655 3703 3920 4255 4258 4265 4640 4666 4795 4817 4838 4867 5004 5173 5233 5236 5689 5806 5832 5913
5939 6006 6074 6182 6200 6208 6323 6512 6622 6935 7039 7094 7153 7156 7190 7438 7567 7602 7681 7742 7774 7814 7905 7957 8160
8275 8518 8521 8534 8624 8696 8786 8807 8809 8882 8947 9003 9174 9175 9199 9261 9379 9438 9530 9547 9801 9948 9994 10176
9990069 9990122 9990161 9990205 9990206 9990213 9990219 9990234 9990270 9990291 9990445 9990504 9990642 9990678 9990
747 9990787 9990950 9990987 9991066 9991204 9991216 9991269 9991370 9991405 9991419 9991425 9991464 9991510 9991637 9991785
9991787 9991794 9991834 9992003 9992118 9992401 9992401 9992542 9992574 9992663 9992895 9993315 9993391 9993395 9993457 9993
548 9993589 9993981 9994005 9994253 9994403 9994490 9994496 9994560 9994645 9994707 9994760 9994824 9994934 9994999 9995012
9995097 9995100 9995196 9995323 9995709 9995973 9996180 9996271 9996316 9996420 9996666 9996697 9997086 9997274 9997536 9997
540 9997737 9997798 9997853 9997920 9997970 9997981 9998000 9998088 9998144 9998157 9998240 9998334 9998412 9998777 9998804
9998952 9998960 9998962 9999212 9999253 9999346 9999482 9999996
```

```

Insertion Sort:
    runtime           : 15.0
    extra memory allocated : 0
    Data:
        5 6 64 122 171 369 401 499 667 794 1005 1281 1414 1822 2076 2342 2460 2545 2616 2620 2674 2752 2877 2932 3070 3341 3
353 3481 3561 3589 3649 3655 3703 3920 4255 4258 4265 4640 4666 4795 4817 4838 4867 5004 5173 5233 5236 5689 5806 5832 5913
5939 6006 6074 6182 6200 6208 6323 6512 6622 6935 7039 7094 7153 7156 7190 7438 7567 7602 7681 7742 7774 7814 7905 7957 8160
    8275 8518 8521 8534 8624 8696 8786 8807 8809 8882 8947 9003 9174 9175 9199 9261 9379 9438 9530 9547 9801 9948 9994 10176
    9990069 9990122 9990161 9990205 9990206 9990213 9990219 9990234 9990270 9990291 9990445 9990504 9990642 9990678 9990
747 9990787 9990950 9990987 9991066 9991204 9991216 9991269 9991370 9991405 9991419 9991425 9991464 9991510 9991637 9991785
9991787 9991794 9991834 9992003 9992118 9992401 9992401 9992542 9992574 9992663 9992895 9993315 9993391 9993395 9993457 9993
548 9993589 9993981 9994005 9994253 9994403 9994490 9994496 9994560 9994645 9994707 9994760 9994824 9994934 9994999 9995012
9995097 9995100 9995196 9995323 9995709 9995973 9996180 9996271 9996316 9996420 9996666 9996697 9997086 9997274 9997536 9997
540 9997737 9997798 9997853 9997920 9997970 9997981 9998000 9998088 9998144 9998157 9998240 9998334 9998412 9998777 9998804
9998952 9998960 9998962 9999212 9999253 9999346 9999482 9999996

Bubble Sort:
    runtime           : 48.1
    extra memory allocated : 0
    Data:
        5 6 64 122 171 369 401 499 667 794 1005 1281 1414 1822 2076 2342 2460 2545 2616 2620 2674 2752 2877 2932 3070 3341 3
353 3481 3561 3589 3649 3655 3703 3920 4255 4258 4265 4640 4666 4795 4817 4838 4867 5004 5173 5233 5236 5689 5806 5832 5913
5939 6006 6074 6182 6200 6208 6323 6512 6622 6935 7039 7094 7153 7156 7190 7438 7567 7602 7681 7742 7774 7814 7905 7957 8160
    8275 8518 8521 8534 8624 8696 8786 8807 8809 8882 8947 9003 9174 9175 9199 9261 9379 9438 9530 9547 9801 9948 9994 10176
    9990069 9990122 9990161 9990205 9990206 9990213 9990219 9990234 9990270 9990291 9990445 9990504 9990642 9990678 9990
747 9990787 9990950 9990987 9991066 9991204 9991216 9991269 9991370 9991405 9991419 9991425 9991464 9991510 9991637 9991785
9991787 9991794 9991834 9992003 9992118 9992401 9992401 9992542 9992574 9992663 9992895 9993315 9993391 9993395 9993457 9993
548 9993589 9993981 9994005 9994253 9994403 9994490 9994496 9994560 9994645 9994707 9994760 9994824 9994934 9994999 9995012
9995097 9995100 9995196 9995323 9995709 9995973 9996180 9996271 9996316 9996420 9996666 9996697 9997086 9997274 9997536 9997
540 9997737 9997798 9997853 9997920 9997970 9997981 9998000 9998088 9998144 9998157 9998240 9998334 9998412 9998777 9998804
9998952 9998960 9998962 9999212 9999253 9999346 9999482 9999996

Merge Sort:
    runtime           : 0.0
    extra memory allocated : 0
    Data:
        5 6 64 122 171 369 401 499 667 794 1005 1281 1414 1822 2076 2342 2460 2545 2616 2620 2674 2752 2877 2932 3070 3341 3
353 3481 3561 3589 3649 3655 3703 3920 4255 4258 4265 4640 4666 4795 4817 4838 4867 5004 5173 5233 5236 5689 5806 5832 5913
5939 6006 6074 6182 6200 6208 6323 6512 6622 6935 7039 7094 7153 7156 7190 7438 7567 7602 7681 7742 7774 7814 7905 7957 8160
    8275 8518 8521 8534 8624 8696 8786 8807 8809 8882 8947 9003 9174 9175 9199 9261 9379 9438 9530 9547 9801 9948 9994 10176
    9990069 9990122 9990161 9990205 9990206 9990213 9990219 9990234 9990270 9990291 9990445 9990504 9990642 9990678 9990
747 9990787 9990950 9990987 9991066 9991204 9991216 9991269 9991370 9991405 9991419 9991425 9991464 9991510 9991637 9991785
9991787 9991794 9991834 9992003 9992118 9992401 9992401 9992542 9992574 9992663 9992895 9993315 9993391 9993395 9993457 9993
548 9993589 9993981 9994005 9994253 9994403 9994490 9994496 9994560 9994645 9994707 9994760 9994824 9994934 9994999 9995012
9995097 9995100 9995196 9995323 9995709 9995973 9996180 9996271 9996316 9996420 9996666 9996697 9997086 9997274 9997536 9997
540 9997737 9997798 9997853 9997920 9997970 9997981 9998000 9998088 9998144 9998157 9998240 9998334 9998412 9998777 9998804
9998952 9998960 9998962 9999212 9999253 9999346 9999482 9999996

Heap Sort:
    runtime           : 0.0
    extra memory allocated : 0
    Data:
        5 6 64 122 171 369 401 499 667 794 1005 1281 1414 1822 2076 2342 2460 2545 2616 2620 2674 2752 2877 2932 3070 3341 3
353 3481 3561 3589 3649 3655 3703 3920 4255 4258 4265 4640 4666 4795 4817 4838 4867 5004 5173 5233 5236 5689 5806 5832 5913
5939 6006 6074 6182 6200 6208 6323 6512 6622 6935 7039 7094 7153 7156 7190 7438 7567 7602 7681 7742 7774 7814 7905 7957 8160
    8275 8518 8521 8534 8624 8696 8786 8807 8809 8882 8947 9003 9174 9175 9199 9261 9379 9438 9530 9547 9801 9948 9994 10176
    9990069 9990122 9990161 9990205 9990206 9990213 9990219 9990234 9990270 9990291 9990445 9990504 9990642 9990678 9990
747 9990787 9990950 9990987 9991066 9991204 9991216 9991269 9991370 9991405 9991419 9991425 9991464 9991510 9991637 9991785
9991787 9991794 9991834 9992003 9992118 9992401 9992401 9992542 9992574 9992663 9992895 9993315 9993391 9993395 9993457 9993
548 9993589 9993981 9994005 9994253 9994403 9994490 9994496 9994560 9994645 9994707 9994760 9994824 9994934 9994999 9995012
9995097 9995100 9995196 9995323 9995709 9995973 9996180 9996271 9996316 9996420 9996666 9996697 9997086 9997274 9997536 9997
540 9997737 9997798 9997853 9997920 9997970 9997981 9998000 9998088 9998144 9998157 9998240 9998334 9998412 9998777 9998804
9998952 9998960 9998962 9999212 9999253 9999346 9999482 9999996

Extra memory deallocated, size: 400000
Extra memory deallocated, size: 400000
Extra memory allocated, size: 4000000
Extra memory allocated, size: 4000000
-----
Dataset Size : 1000000
-----
Selection Sort:

```

Summary of comparison of sorting algorithms:

- Selection Sort

Selection sort is a simple comparison-based algorithm. It divides the input list into two parts: a sorted sublist of items which is built up from left to right at the front (left) of the list, and an unsorted sublist occupying the rest of the list. On each iteration, the algorithm selects the smallest (or largest, depending on sorting order) element from the unsorted sublist, swaps it with the leftmost unsorted element, and moves the sublist boundaries one element to the right. It has a time complexity of $O(n^2)$.

- Insertion Sort

Insertion sort is a simple sorting algorithm that builds the final sorted array (or list) one item at a time. It is much less efficient on large lists than more advanced algorithms like quicksort, merge sort. However, it has advantages: simple implementation, efficient for (small) datasets, more efficient in practice than most other simple quadratic algorithms such as selection sort, and is stable. It works well for data that is already mostly sorted. For worst-case scenario, the complexity is $O(n^2)$. For best case scenario, the complexity is $O(n)$.

- Bubble Sort

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted. The algorithm gets its name from the way smaller elements "bubble" to the top of the list (beginning of the list). Like insertion and selection sorts, bubble sort has a time complexity of $O(n^2)$.

- Merge Sort

Merge sort is a divide-and-conquer algorithm. It divides the unsorted list into N sublists, each containing one element (a list of one element is considered sorted), then repeatedly merges sublists to produce new sorted sublists until there is only one sublist remaining. This will be the sorted list. It is much more efficient than the simple algorithms mentioned above, with a time complexity of $O(n \log n)$.

- Heap Sort

Heap sort is a comparison-based sorting technique based on a binary heap data structure. It is similar to selection sort where we first find the maximum element and place the maximum element at the end. We repeat the same process for the remaining elements. The time complexity of heap sort is $O(n \log n)$ in all cases, and it has the advantage of not using any extra storage (apart from variable storage), unlike merge sort.