

COMP 2131

Output Capture of Assignment 1

Author : Koki Yamanaka

Student ID : T00681865

Submission date : 21st Oct 2022

Question 1 (Shell Sort) output capture :

Test case 1: given sample array

Test case 1

original unsorted array : [9, 6, 8, 12, 3, 1, 7]

6 and 3 are not in order,so swap
after swap:[9, 3, 8, 12, 6, 1, 7]
8 and 1 are not in order,so swap
after swap:[9, 3, 1, 12, 6, 8, 7]
12 and 7 are not in order,so swap
after swap:[9, 3, 1, 7, 6, 8, 12]
9 and 7 are not in order,so swap
after swap:[7, 3, 1, 9, 6, 8, 12]
7 and 3 are not in order,so swap
after swap:[3, 7, 1, 9, 6, 8, 12]
7 and 1 are not in order,so swap
after swap:[3, 1, 7, 9, 6, 8, 12]
9 and 6 are not in order,so swap
after swap:[3, 1, 7, 6, 9, 8, 12]
9 and 8 are not in order,so swap
after swap:[3, 1, 7, 6, 8, 9, 12]
3 and 1 are not in order,so swap
after swap:[1, 3, 7, 6, 8, 9, 12]
7 and 6 are not in order,so swap
after swap:[1, 3, 6, 7, 8, 9, 12]

final sorted array :[1, 3, 6, 7, 8, 9, 12]

Test case 2 : 10 random integers

Test case 2

original unsorted array : [114, 377, 305, 137, 180, 370, 222, 70, 242, 249]

377 and 222 are not in order,so swap

after swap:[114, 222, 305, 137, 180, 370, 377, 70, 242, 249]

305 and 70 are not in order,so swap

after swap:[114, 222, 70, 137, 180, 370, 377, 305, 242, 249]

114 and 70 are not in order,so swap

after swap:[70, 222, 114, 137, 180, 370, 377, 305, 242, 249]

222 and 137 are not in order,so swap

after swap:[70, 137, 114, 222, 180, 370, 377, 305, 242, 249]

370 and 305 are not in order,so swap

after swap:[70, 137, 114, 222, 180, 305, 377, 370, 242, 249]

377 and 242 are not in order,so swap

after swap:[70, 137, 114, 222, 180, 305, 242, 370, 377, 249]

370 and 249 are not in order,so swap

after swap:[70, 137, 114, 222, 180, 305, 242, 249, 377, 370]

305 and 249 are not in order,so swap

after swap:[70, 137, 114, 222, 180, 249, 242, 305, 377, 370]

137 and 114 are not in order,so swap

after swap:[70, 114, 137, 222, 180, 249, 242, 305, 377, 370]

222 and 180 are not in order,so swap

after swap:[70, 114, 137, 180, 222, 249, 242, 305, 377, 370]

249 and 242 are not in order,so swap

after swap:[70, 114, 137, 180, 222, 242, 249, 305, 377, 370]

377 and 370 are not in order,so swap

after swap:[70, 114, 137, 180, 222, 242, 249, 305, 370, 377]

final sorted array[70, 114, 137, 180, 222, 242, 249, 305, 370, 377]

Test case 3: 20 random integers

Test case 3

original unsorted array : [328, 431, 558, 383, 511, 142, 543, 204, 69, 405, 79, 483, 15, 273, 374, 506, 478, 314, 597, 595]

328 and 79 are not in order,so swap

after swap:[79, 431, 558, 383, 511, 142, 543, 204, 69, 405, 328, 483, 15, 273, 374, 506, 478, 314, 597, 595]

558 and 15 are not in order,so swap

after swap:[79, 431, 15, 383, 511, 142, 543, 204, 69, 405, 328, 483, 558, 273, 374, 506, 478, 314, 597, 595]

383 and 273 are not in order,so swap

after swap:[79, 431, 15, 273, 511, 142, 543, 204, 69, 405, 328, 483, 558, 383, 374, 506, 478, 314, 597, 595]

511 and 374 are not in order,so swap

after swap:[79, 431, 15, 273, 374, 142, 543, 204, 69, 405, 328, 483, 558, 383, 511, 506, 478, 314, 597, 595]

543 and 478 are not in order,so swap

after swap:[79, 431, 15, 273, 374, 142, 478, 204, 69, 405, 328, 483, 558, 383, 511, 506, 543, 314, 597, 595]

273 and 69 are not in order,so swap

after swap:[79, 431, 15, 69, 374, 142, 478, 204, 273, 405, 328, 483, 558, 383, 511, 506, 543, 314, 597, 595]

558 and 314 are not in order,so swap

after swap:[79, 431, 15, 69, 374, 142, 478, 204, 273, 405, 328, 483, 314, 383, 511, 506, 543, 558, 597, 595]

79 and 15 are not in order,so swap

after swap:[15, 431, 79, 69, 374, 142, 478, 204, 273, 405, 328, 483, 314, 383, 511, 506, 543, 558, 597, 595]

431 and 69 are not in order,so swap

after swap:[15, 69, 79, 431, 374, 142, 478, 204, 273, 405, 328, 483, 314, 383, 511, 506, 543, 558, 597, 595]

431 and 142 are not in order,so swap

after swap:[15, 69, 79, 142, 374, 431, 478, 204, 273, 405, 328, 483, 314, 383, 511, 506, 543, 558, 597, 595]

431 and 204 are not in order,so swap

after swap:[15, 69, 79, 142, 374, 204, 478, 431, 273, 405, 328, 483, 314, 383, 511, 506, 543, 558, 597, 595]

478 and 273 are not in order,so swap

after swap:[15, 69, 79, 142, 374, 204, 273, 431, 478, 405, 328, 483, 314, 383, 511, 506, 543, 558, 597, 595]

Continue

431 and 405 are not in order,so swap

after swap:[15, 69, 79, 142, 374, 204, 273, 405, 478, 431, 328, 483, 314, 383, 511, 506, 543, 558, 597, 595]

478 and 328 are not in order,so swap

after swap:[15, 69, 79, 142, 374, 204, 273, 405, 328, 431, 478, 483, 314, 383, 511, 506, 543, 558, 597, 595]

478 and 314 are not in order,so swap

after swap:[15, 69, 79, 142, 374, 204, 273, 405, 328, 431, 314, 483, 478, 383, 511, 506, 543, 558, 597, 595]

483 and 383 are not in order,so swap

after swap:[15, 69, 79, 142, 374, 204, 273, 405, 328, 431, 314, 383, 478, 483, 511, 506, 543, 558, 597, 595]

374 and 273 are not in order,so swap

after swap:[15, 69, 79, 142, 273, 204, 374, 405, 328, 431, 314, 383, 478, 483, 511, 506, 543, 558, 597, 595]

374 and 328 are not in order,so swap

after swap:[15, 69, 79, 142, 273, 204, 328, 405, 374, 431, 314, 383, 478, 483, 511, 506, 543, 558, 597, 595]

374 and 314 are not in order,so swap

after swap:[15, 69, 79, 142, 273, 204, 328, 405, 314, 431, 374, 383, 478, 483, 511, 506, 543, 558, 597, 595]

431 and 383 are not in order,so swap

after swap:[15, 69, 79, 142, 273, 204, 328, 405, 314, 383, 374, 431, 478, 483, 511, 506, 543, 558, 597, 595]

328 and 314 are not in order,so swap

after swap:[15, 69, 79, 142, 273, 204, 314, 405, 328, 383, 374, 431, 478, 483, 511, 506, 543, 558, 597, 595]

405 and 383 are not in order,so swap

after swap:[15, 69, 79, 142, 273, 204, 314, 383, 328, 405, 374, 431, 478, 483, 511, 506, 543, 558, 597, 595]

273 and 204 are not in order,so swap

after swap:[15, 69, 79, 142, 204, 273, 314, 383, 328, 405, 374, 431, 478, 483, 511, 506, 543, 558, 597, 595]

383 and 328 are not in order,so swap

after swap:[15, 69, 79, 142, 204, 273, 314, 328, 383, 405, 374, 431, 478, 483, 511, 506, 543, 558, 597, 595]

405 and 374 are not in order,so swap

after swap:[15, 69, 79, 142, 204, 273, 314, 328, 383, 374, 405, 431, 478, 483, 511, 506, 543, 558, 597, 595]

Continue

511 and 506 are not in order,so swap

after swap:[15, 69, 79, 142, 204, 273, 314, 328, 383, 374, 405, 431, 478, 483, 506, 511, 543, 558, 597, 595]

597 and 595 are not in order,so swap

after swap:[15, 69, 79, 142, 204, 273, 314, 328, 383, 374, 405, 431, 478, 483, 506, 511, 543, 558, 595, 597]

383 and 374 are not in order,so swap

after swap:[15, 69, 79, 142, 204, 273, 314, 328, 374, 383, 405, 431, 478, 483, 506, 511, 543, 558, 595, 597]

final sorted array[15, 69, 79, 142, 204, 273, 314, 328, 374, 383, 405, 431, 478, 483, 506, 511, 543, 558, 595, 597]

Question 2 (Efficient Bubble Sort)

Test case 1 : array with unsorted random integers

Blue! Terminal Window - EfficientBubbleSort

Options

```
TEST1 - original bubble sort with unsorted array
before sorted : [6, 52, 160, 167, 286, 216, 56, 166, 374, 280]
current pass is 1 : [6, 52, 160, 167, 286, 216, 56, 166, 374, 280]
current pass is 2 : [6, 52, 160, 167, 216, 56, 166, 286, 280, 374]
current pass is 3 : [6, 52, 160, 167, 56, 166, 216, 280, 286, 374]
current pass is 4 : [6, 52, 160, 56, 166, 167, 216, 280, 286, 374]
current pass is 5 : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
current pass is 6 : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
current pass is 7 : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
current pass is 8 : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
current pass is 9 : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
current pass is 10 : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
after sorted : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
```

```
-----
TEST1 - modified bubble sort with unsorted array
before sorted : [6, 52, 160, 167, 286, 216, 56, 166, 374, 280]
current pass is 1 : [6, 52, 160, 167, 286, 216, 56, 166, 374, 280]
current pass is 2 : [6, 52, 160, 167, 216, 56, 166, 286, 280, 374]
current pass is 3 : [6, 52, 160, 167, 56, 166, 216, 280, 286, 374]
current pass is 4 : [6, 52, 160, 56, 166, 167, 216, 280, 286, 374]
current pass is 5 : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
after sorted : [6, 52, 56, 160, 166, 167, 216, 280, 286, 374]
```

Test case 2 : array with sorted random integers

```
TEST2 - original bubble sort with sorted array
before sorting : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 1 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 2 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 3 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 4 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 5 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 6 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 7 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 8 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 9 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 10 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
after sorting : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
```

```
-----
TEST2 - modified bubble sort with sorted array
before sorting : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
current pass is 1 : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
after sorting : [3, 119, 120, 400, 451, 477, 900, 1000, 1232, 2444]
```

Question 3 (Algorithm analysis)

Test case 1 : array with the size of 10

TEST CASE 1 - SIZE 10 ARRAY

=====

SHELL SORT

<unsorted>

Total comparisons made : 53

Total swaps made : 12

Execution time in milliseconds: 0.0208

<sorted>

Total comparisons made : 22

Total swaps made : 0

Execution time in milliseconds: 0.0036

BUBBLE SORT

<unsorted>

Total comparisons made : 45

Total swaps made : 30

Execution time in milliseconds: 0.0078

<sorted>

Total comparisons made : 45

Total swaps made : 0

Execution time in milliseconds: 0.0051

BUBBLE SORT 2

<unsorted>total number comparison made : 45

TOTAL number swaps made : 30

Execution time in milliseconds: 0.0076

<sorted>total number comparison made : 9

TOTAL number swaps made : 0

Execution time in milliseconds: 0.0019

Test case 2 : array with the size of 100

TEST CASE 2 - SIZE 100 ARRAY

=====

SHELL SORT

<unsorted>

Total comparisons made : 2808

Total swaps made : 395

Execution time in milliseconds: 0.1978

<sorted>

Total comparisons made : 503

Total swaps made : 0

Execution time in milliseconds: 0.0893

BUBBLE SORT

<unsorted>

Total comparisons made : 4950

Total swaps made : 2382

Execution time in milliseconds: 0.3185

<sorted>

Total comparisons made : 4950

Total swaps made : 0

Execution time in milliseconds: 0.184

BUBBLE SORT 2

<unsorted>total number comparison made : 4905

TOTAL number swaps made : 2382

Execution time in milliseconds: 0.9358

<sorted>total number comparison made : 99

TOTAL number swaps made : 0

Execution time in milliseconds: 0.0067

Test case 3 : array with the size of 1000

TEST CASE 3 - SIZE 1000 ARRAY

=====

SHELL SORT

<unsorted>

Total comparisons made : 55727

Total swaps made : 7455

Execution time in milliseconds: 4.5337

<sorted>

Total comparisons made : 8006

Total swaps made : 0

Execution time in milliseconds: 0.9166

BUBBLE SORT

<unsorted>

Total comparisons made : 499500

Total swaps made : 260612

Execution time in milliseconds: 27.3134

<sorted>

Total comparisons made : 499500

Total swaps made : 0

Execution time in milliseconds: 13.2808

BUBBLE SORT 2

<unsorted>total number comparison made : 499065

TOTAL number swaps made : 260612

Execution time in milliseconds: 24.49

<sorted>total number comparison made : 999

TOTAL number swaps made : 0

Execution time in milliseconds: 0.0939
