Exercises for lesson 5: Priority queues and sorting

**Exercise 5.1**

Which of the following statements are true?

|  |  |
| --- | --- |
| 1. | BubbleSort is a recursive algorithm. **false** |
|  |  |

|  |  |
| --- | --- |
| 2. | InsertionSort is considered a quadratic sort. **true** |

|  |  |
| --- | --- |
| 3. | The improvement of MergeSort over InsertionSort is much more significant for small arrays. **false** |
|  |  |

**Exercise 5.2**

Fill out the blanks:

|  |  |
| --- | --- |
| 1. | \_\_\_\_\_\_\_\_Insertion\_\_\_\_\_\_\_\_\_\_\_\_ sorts an array by making several passes through the array, selecting the next smallest item each time and placing it where it belongs in the array. |

|  |  |
| --- | --- |
| 2. | In the best case, InsertionSort makes O(\_\_n\_\_\_\_) comparisons. |

|  |  |
| --- | --- |
| 3. | With respect to MergeSort, additional space usage is O(\_\_\_\_\_\_\_n\_\_\_\_\_\_\_\_\_\_\_\_). |

|  |  |
| --- | --- |
| 4. | The following is the \_\_\_merge\_\_\_\_\_\_\_\_\_\_\_ algorithm.    Access the first item from both sequences.  while not finished with either sequence       Compare the current items from the two sequences, copy the smaller current item to the output sequence, and       access the next item from the input sequence whose item was copied  Copy any remaining items from the first sequence to the output sequence.  Copy any remaining items from the second sequence to the output sequence. |

|  |  |
| --- | --- |
| 5. | In MergeSort, the total effort to reconstruct the sorted array through merging is \_\_c)\_\_. |
| A) | O(1) |
| B) | O(log n) |
| C) | O(n log n) |
| D) | O(n2) |

|  |  |
| --- | --- |
| 6. | Which of the following sorts is not O(n log(n))?  A) |
| A) | Insertion |
| B) | Heap |
| C) | Merge |
|  |  |

|  |  |
| --- | --- |
| 7. | The best general-purpose sorting algorithms provide \_\_D)\_\_ average-case behavior and are considerably faster for large arrays. |
| A) | O(1) |
| B) | O(n) |
| C) | O(n2) |
| D) | O(n log n) |

|  |  |
| --- | --- |
| 8. | Which of the following generally gives the worst performance? B) |
| A) | HeapSort |
| B) | BubbleSort |
| C) | BucketSort |
| D) | MergeSort |

|  |  |
| --- | --- |
| 9. | The following represents the algorithm for \_\_Insertion\_\_\_\_\_Sort.    for each array element from the second (nextPos = 1) to the last     Insert the element at nextPos where it belongs in the array, increasing     the length of the sorted subarray by 1 element. |

**Exercise 5.3**

Tear a piece of paper into *n* pieces and write a number on each piece of paper – or grab a stack of playing cards – put them in random order and sort them “physically” using BubbleSort, InsertionSort, MergeSort, QuickSort, HeapSort and BucketSort.

**Exercise 5.4**

Complete the implementation of a min-heap in java.