Each row in the table represents an iteration of the outer loop or recursive call (depending on the sorting algorithm). Each column within a row represents a different spot in the array that is being sorted. Fill in the table as the specific sorting algorithm would change the array. If a value in a column does not change, don’t write it over again in the next row. You may not need all the rows. The bold row represents the initial values in the array. Use the algorithms as we discussed in class.

1. Quick sort using the first item in the current section as the pivot. Highlight (color) the pivots after they have been put in their final location (in the first call the 5 will be the pivot). Only highlight the pivot on the line when it was used as a pivot. Show the 2 sides of a recursion on different lines so I can more easily read your answers.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **3** | **7** | **6** | **4** | **1** | **8** | **2** | **9** | **0** |
|  |  | 0 |  |  |  |  |  |  | 7 |
|  |  |  | 2 |  |  |  | 6 |  |  |
| 1 |  |  |  |  | 5 |  |  |  |  |
|  | 0 | 3 |  |  |  |  |  |  |  |
| 0 | 1 |  |  |  |  |  |  |  |  |
|  |  | 2 | 3 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 7 | 9 |
|  |  |  |  |  |  | 7 |  | 8 |  |
|  |  |  |  |  |  | 6 | 7 |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

1. Merge. When halving an odd number, truncate as C++ would do. For example, on the second level when you have 5 elements, you should split them such that there are 2 on the left and 3 on the right. Show the 2 sides of a recursion on different lines so I can more easily read your answers.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **3** | **7** | **6** | **4** | **1** | **8** | **2** | **9** | **0** |
| 3 | 5 |  |  |  |  |  |  |  |  |
|  |  |  | 4 | 6 |  |  |  |  |  |
|  |  | 4 | 6 | 7 |  |  |  |  |  |
|  | 4 | 5 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 0 | 9 |
|  |  |  |  |  |  |  | 0 | 2 |  |
|  |  |  |  |  | 0 | 1 | 2 | 8 |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |