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. Selection (finding the minimum). The first iteration is done for you. The 0 and 5 swapped and the rest of the values remained the same. If nothing changes in an iteration, leave a blank line.

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

1. Insertion (linear and binary make the same changes to the array). If nothing changes in an iteration, then leave a blank line.

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