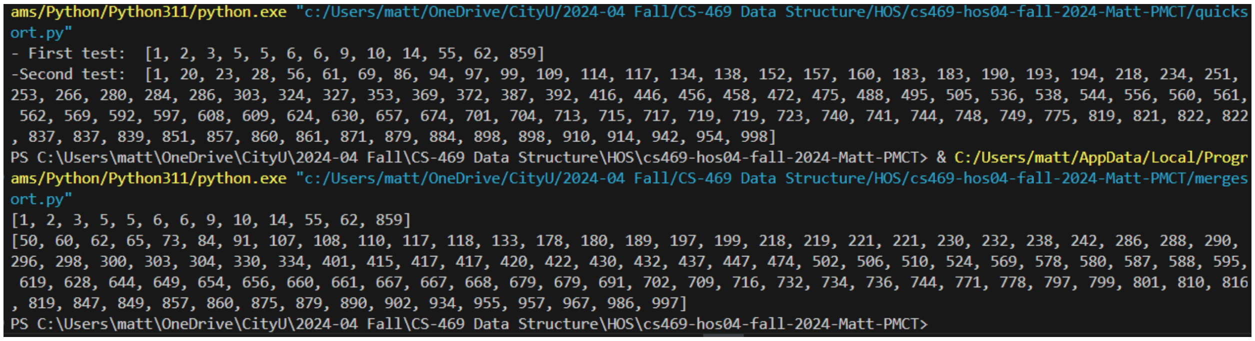

```
ams/Python/Python311/python.exe "c:/Users/matt/OneDrive/CityU/2024-04 Fall/CS-469 Data Structure/HOS/cs469-hos04-fall-2024-Matt-PMCT/quicks
ort.py"
- First test:  [1, 2, 3, 5, 5, 6, 6, 9, 10, 14, 55, 62, 859]
-Second test:  [1, 20, 23, 28, 56, 61, 69, 86, 94, 97, 99, 109, 114, 117, 134, 138, 152, 157, 160, 183, 183, 190, 193, 194, 218, 234, 251,
253, 266, 280, 284, 286, 303, 324, 327, 353, 369, 372, 387, 392, 416, 446, 456, 458, 472, 475, 488, 495, 505, 536, 538, 544, 556, 560, 561,
 562, 569, 592, 597, 608, 609, 624, 630, 657, 674, 701, 704, 713, 715, 717, 719, 719, 723, 740, 741, 744, 748, 749, 775, 819, 821, 822, 822
, 837, 837, 839, 851, 857, 860, 861, 871, 879, 884, 898, 898, 910, 914, 942, 954, 998]
PS C:\Users\matt\OneDrive\CityU\2024-04 Fall\CS-469 Data Structure\HOS\cs469-hos04-fall-2024-Matt-PMCT> & C:/Users/matt/AppData/Local/Progr
ams/Python/Python311/python.exe "c:/Users/matt/OneDrive/CityU/2024-04 Fall/CS-469 Data Structure/HOS/cs469-hos04-fall-2024-Matt-PMCT/merges
ort.py"
[1, 2, 3, 5, 5, 6, 6, 9, 10, 14, 55, 62, 859]
[50, 60, 62, 65, 73, 84, 91, 107, 108, 110, 117, 118, 133, 178, 180, 189, 197, 199, 218, 219, 221, 221, 230, 232, 238, 242, 286, 288, 290,
296, 298, 300, 303, 304, 330, 334, 401, 415, 417, 417, 420, 422, 430, 432, 437, 447, 474, 502, 506, 510, 524, 569, 578, 580, 587, 588, 595,
 619, 628, 644, 649, 654, 656, 660, 661, 667, 667, 668, 679, 679, 691, 702, 709, 716, 732, 734, 736, 744, 771, 778, 797, 799, 801, 810, 816
, 819, 847, 849, 857, 860, 875, 879, 890, 902, 934, 955, 957, 967, 986, 997]
PS C:\Users\matt\OneDrive\CityU\2024-04 Fall\CS-469 Data Structure\HOS\cs469-hos04-fall-2024-Matt-PMCT>
```

The main difference between Quick Sort and Merge Sort is how they divide. Quick Sort is a partitioning sort that uses a pivot so that items less than the pivot are on one side and items greater than the pivot are on the other side. This lets the sorting be done in place. Merge Sort is a merging sort that divides the arrays into two halves and then recursively sorts each half and merges them back together. This requires more space to store the data while it is being sorted.