By restring the range of the random integers to 10 (or any number less than 2000) and setting the array size to 2000, there will always be repeated elements in the array (by Pigeon Hole Principle). As such, the stability of the arrays can be determined.  
Sorter A is able to sort the array and is unstable.  
Sorter B is able to sort the array and is stable.  
  
Sorter C is able to sort the array for small array sizes but gives unsorted array for large array size (I got unsorted array for n = 40000).  
Similarly, C is stable for small array size but unstable for large array size. It can be concluded that C is Dr. Evil.  
  
Sorter D is able to sort the array and is unstable.  
Sorter E is able to sort the array and is stable.  
Sorters B and E take approximately the same amount of time (0.06s) to sort the given array of size 2000.  
However, for a reversely sorted array (worst case scenario) with a larger size of 40,000, B takes about 16s while E takes about 0.4s to sort. It can be inferred that E has a smaller asymptotic running time compared to B and it is likely to be merge sort (O(n log n)) while B is likely to be insertion sort(O(n^2)).  
  
Sorter F is able to sort the array and is stable.  
F takes a significantly longer time to sort the array input even for small array size of 2000. To illustrate, sorters A to E takes less than 1 sec on average to sort the given array while F takes about 10s to sort the array. Therefore, F is likely to be bubble sort which has the slowest asymptotic running time in practical.  
  
The only unstable algorithms which we have learnt are selection sort and quick sort.  
Sorter A takes a longer time to sort a smaller array of size 2000 than sorter D.  
However, comparing the time taken for sorters A and D to sort a larger random array size of 40,000, A (1.5s) takes a considerably shorter time (approximately 3 times) than D (4s).

In addition, the percentage increase in time taken for sorting from size 2000 to size 40,000 is much larger for D (~33,000%) than A(~500%).  
This shows that the time taken for A increases at a slower rate and hence a smaller asymptotic running time than D. Therefore, A is quick sort (O(n log n) for average case) while D is selection sort (O(n^2)).