

1. C. This sorting algorithm uses a Big Oh Notation of $O(nk)$. Where n is the number of times that the while loop complies. The while loop complies based on the digits of the max number in the array. For example a number 300 will run 3 times total because it is in the hundreds digit. The worst time would $O(k(n+b))$. Where b is the is the number of loops that the four for loops gets repeated and k is the number of times that the while loop gets called. This is the worst case possible in this code if all the numbers ended with the same digits and it's a digit of 9. This will require every alliteration possible to complete the while loop since all the for loops will be called the maximum amount of times.

```

while(max/digit != 0) {
    int[] counter = new int[10];
    // this will hold all the digits into an array
    for(int i = 0; i<arraySize; i++) {
        counter[(hold[i] / digit) % 10]++;
        //adds the digits to the counter
    }
    for(int i = 1; i<10; i++) {
        counter[i] = counter[i] + counter[i-1];
        //idea is similar to the counting sort
    }
    for(int i = arraySize -1; i>= 0; i--) {
        newSort[--counter[(hold[i]/digit)%10]] = hold[i];
        //moves all the counted array at that spot into the newSort
    }
    for(int i = 0; i < arraySize; i++) {
        a[i] = newSort[i];
        //sorts the new sort base on digits into a holder array
    }
    digit *= 10;
    //ones place, tens, hundreds
}

```

1. D. This code is stable because if the code has a repeated number, for example a number like 75 and 25. The digit 5 gets placed where 75 will come before the 2 after the first digit loop is called. Thus this means if the number are the same, for example there are two number 25 in the array, the first 25 that appeared will also be the first 25 that appears in the final sorted array.

```

for(int i = 0; i<arraySize; i++) {
    counter[(hold[i] / digit) % 10]++;
    //adds the digits to the counter
}
for(int i = 1; i<10; i++) {
    counter[i] = counter[i] + counter[i-1];
    //idea is similar to the counting sort
}
for(int i = arraySize -1; i>= 0; i--) {
    newSort[--counter[(hold[i]/digit)%10]] = hold[i];
    //moves all the counted array at that spot into the newSort
}

```