

Assignment 3 Output Example (timings are not correct):

```
> javac MySorts.java  
> java MySorts
```

Unsorted integers:

384 298 748 1832 47 12 948 357 90

Sorting via Comb Sort:

Gap 7: 357 90 748 1832 47 12 948 384 298

Gap 5: 12 90 384 298 47 357 948 748 1832

Gap 4: 12 90 384 298 47 357 948 748 1832

Gap 3: 12 47 357 298 90 384 948 748 1832

Gap 2: 12 47 90 298 357 384 948 748 1832

Gap 1: 12 47 90 298 357 384 748 948 1832

Done - complete in 240ns

Sorting via Radix Sort:

1s - 90 1832 12 384 47 357 298 748 948

10s - 12 1832 47 748 948 357 384 90 298

100s - 12 47 90 298 357 384 748 948 1832

1000s - 12 47 90 298 357 384 748 948 1832

Done - complete in 120ns

Sorting via Snake Sort:

Original:

384 298 748

1832 47 12

948 357 90

Row Pass 1:

298 384 748

1832 47 12

90 357 948

Column Pass 1:

90 47 12

298 357 748

1832 384 948

Row Pass 2:

12 47 90

748 357 298

384 948 1832

Column Pass 2:
12 47 90
384 357 298
748 948 1832

Row Pass 3:
12 47 90
384 357 298
748 948 1832

Column Pass 3:
12 47 90
384 357 298
748 948 1832

Final Pass:
12 47 90
298 357 384
748 948 1832

Done - complete in 88ns

Sorting via BogoSort:
Shuffle 1 - 384 47 357 948 90 298 748 12 1832
Shuffle 2 - 47 12 748 948 298 384 357 90 1832
Shuffle 3 - 1832 384 748 357 90 948 298 47 12
Shuffle 4 - 948 90 1832 357 47 298 748 12 384
Shuffle 5 - 384 948 357 298 47 90 12 748 1832
Shuffle 6 - 298 47 1832 748 357 90 948 12 384
Shuffle 7 - 47 1832 12 298 938 357 384 90 748
Shuffle 8 - 12 47 90 298 357 384 748 948 1832
Done - complete in 3894ns