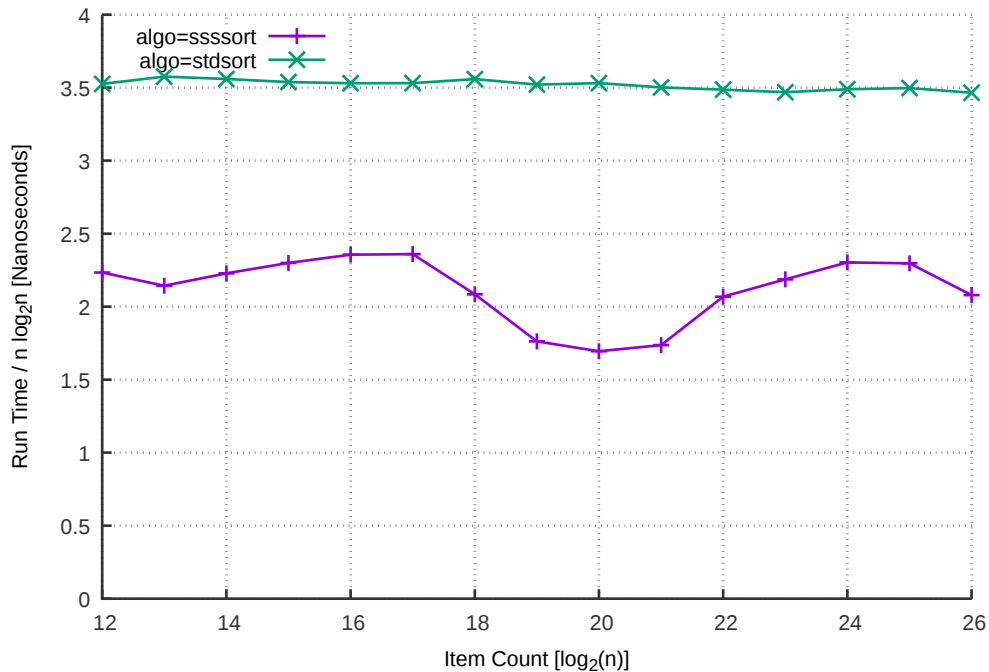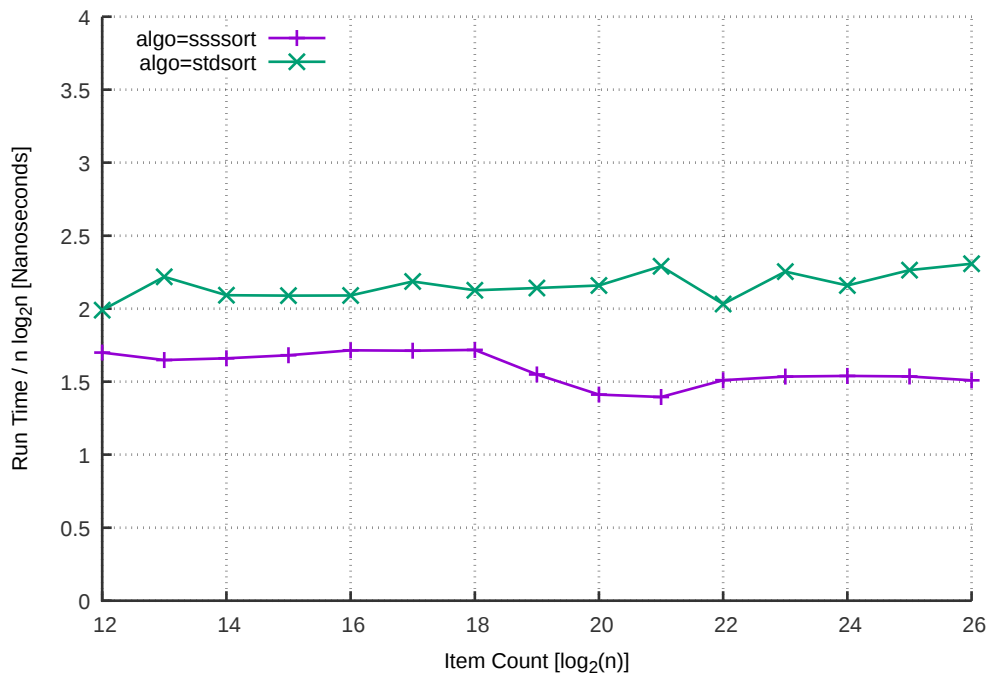Super Scalar Sample Sort Test: Random
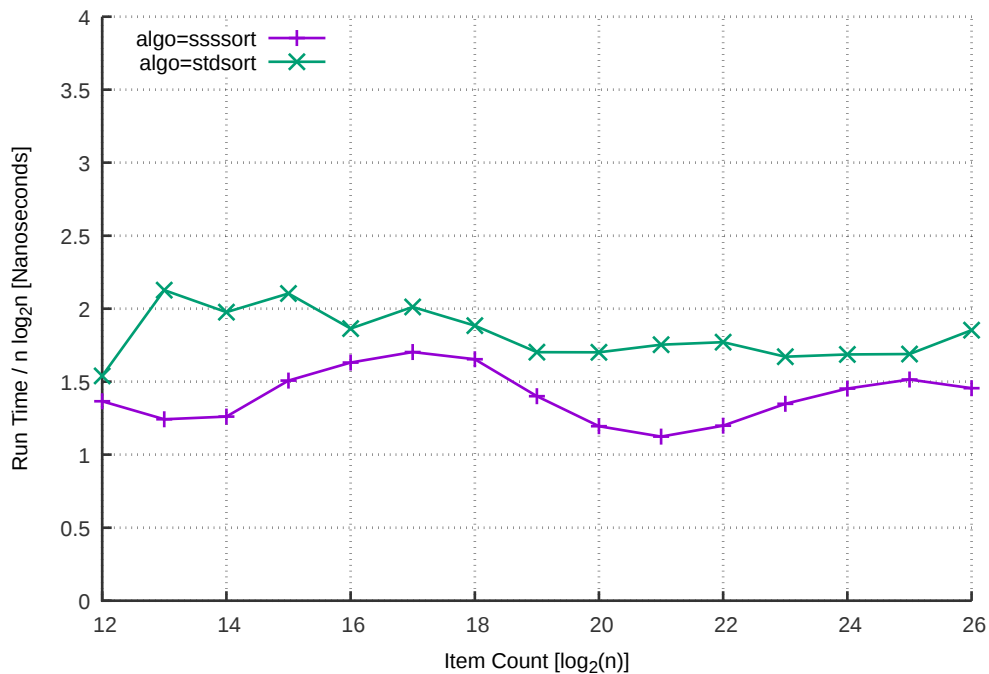
Super Scalar Sample Sort Test: 80% Sorted

algo=ssssort
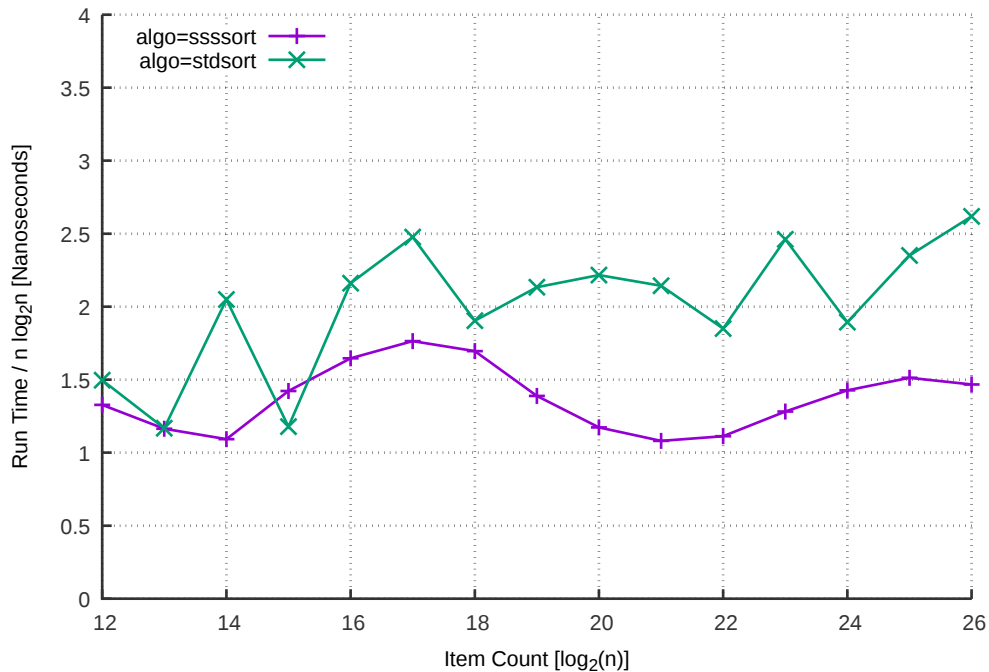algo=stdsort

Run Time / n log$_2$n [Nanoseconds]

Item Count [log$_2$(n)]

Super Scalar Sample Sort Test: 90% Sorted

algo=ssssort
algo=stdsort

Run Time / n $\log_2 n$ [Nanoseconds]

Item Count [$\log_2(n)$]

Super Scalar Sample Sort Test: 99% Sorted

algo=ssssort
algo=stdsort

Run Time / n log$_2$n [Nanoseconds]

Item Count [log$_2$(n)]

Super Scalar Sample Sort Test: 99.9% Sorted

algo=ssssort
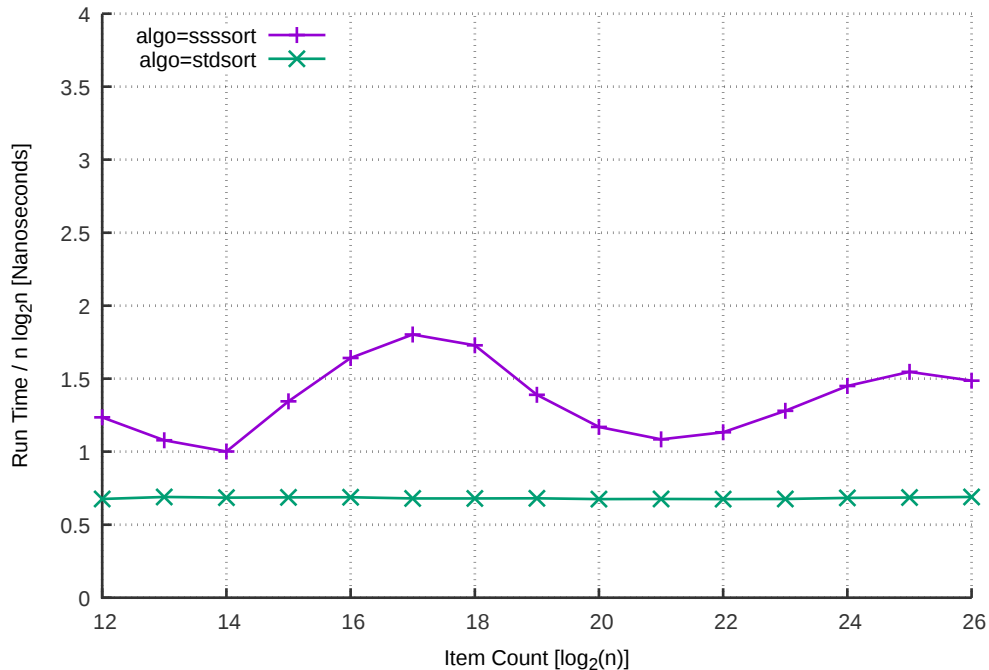algo=stdsort

Run Time / n log₂n [Nanoseconds]

Item Count [log₂(n)]

Super Scalar Sample Sort Test: 90% Sorted + 10% Random Tail
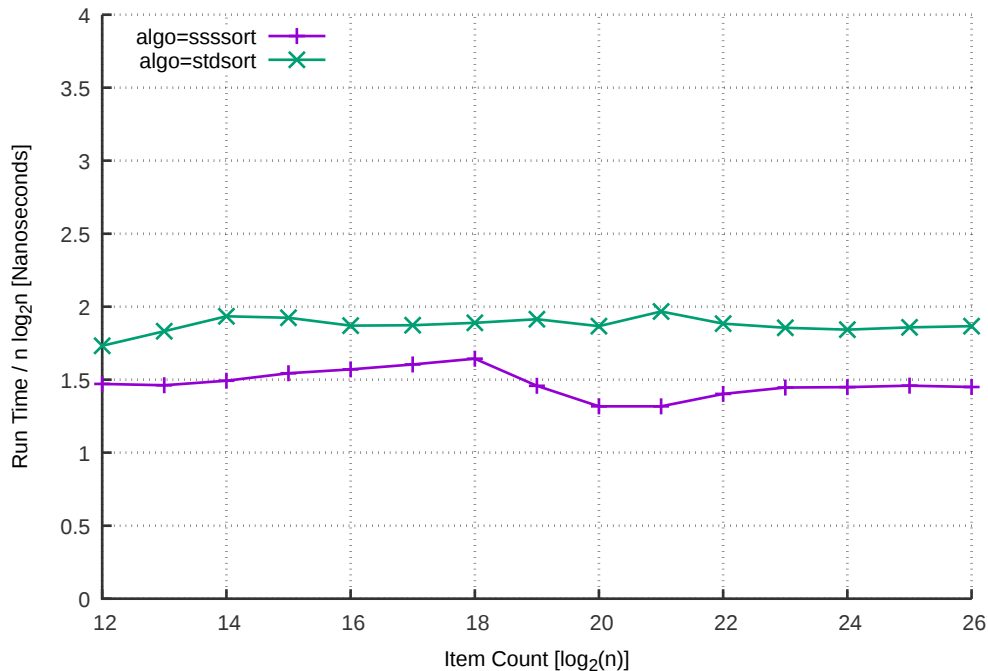
Super Scalar Sample Sort Test: 99% Sorted + 1% Random Tail

algo=ssssort
algo=stdsort
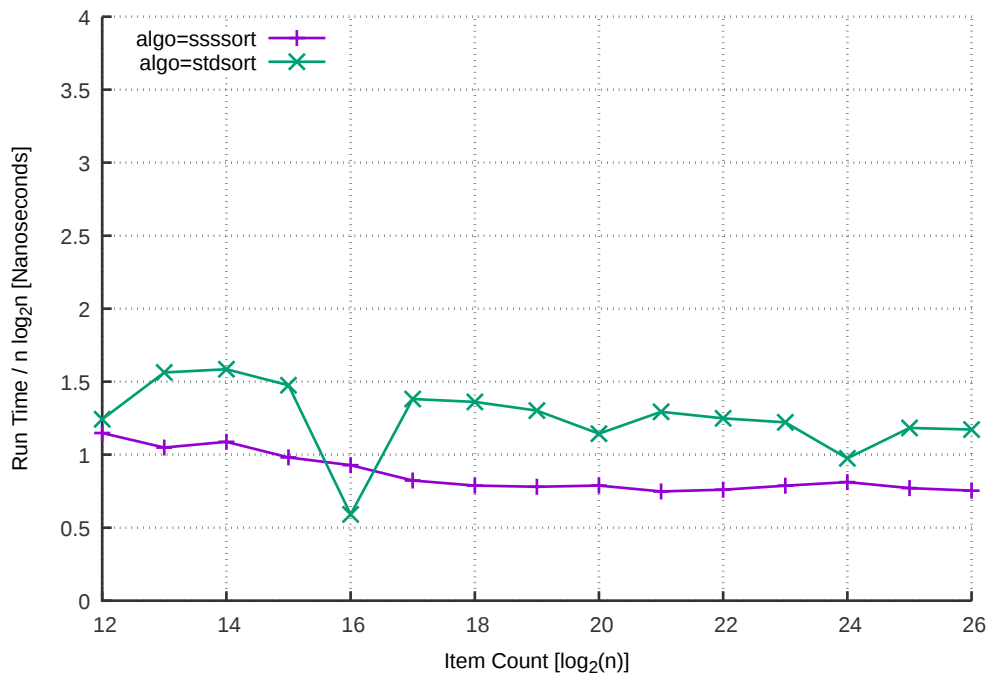
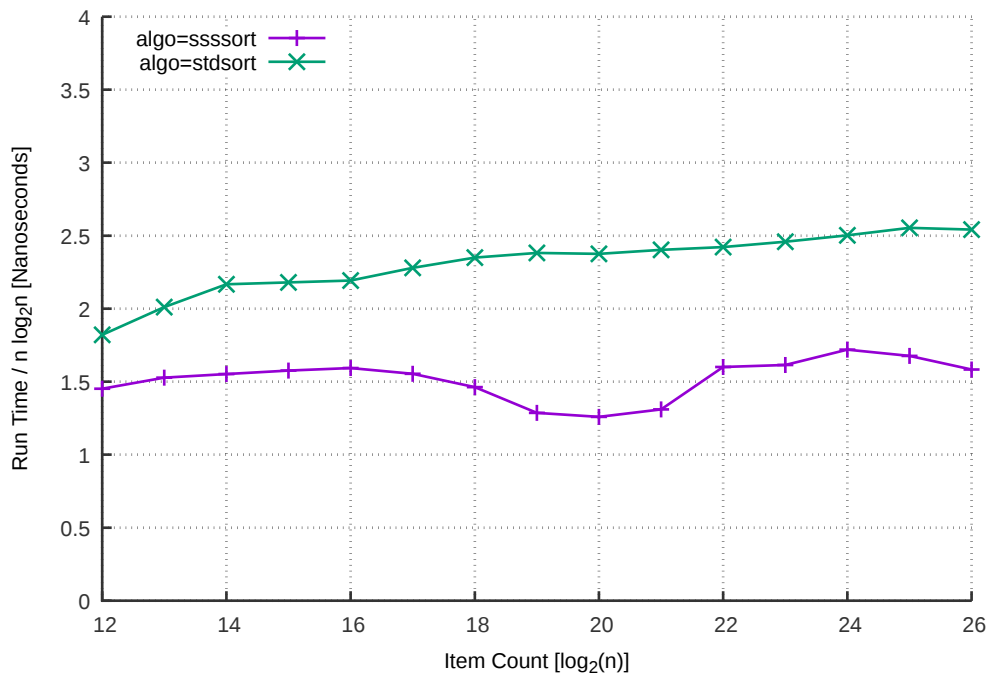Run Time / n $\log_2 n$ [Nanoseconds]

Item Count [$\log_2(n)$]
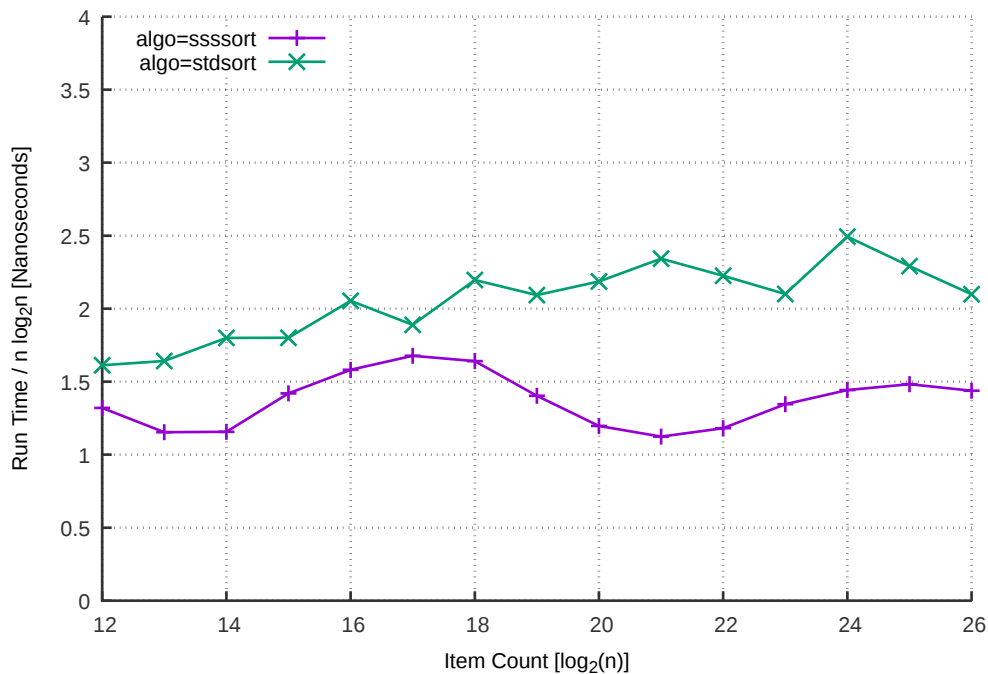
Super Scalar Sample Sort Test: Sorted

Super Scalar Sample Sort Test: Reverse Sorted

Run Time / n log$_2$n [Nanoseconds] vs Item Count [log$_2$(n)]

- algo=ssssort
- algo=stdsort
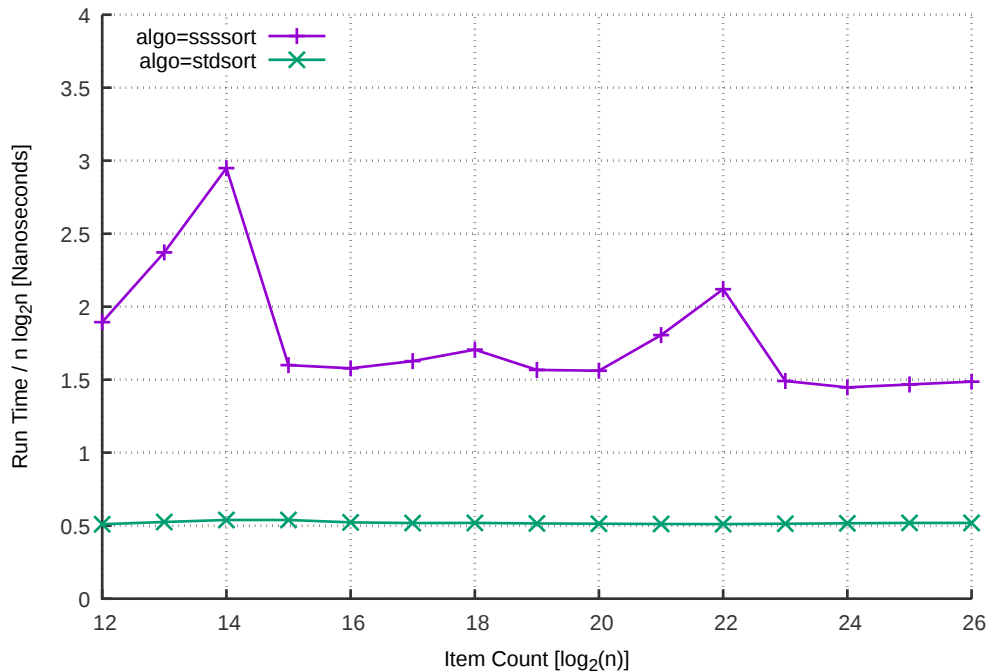
Super Scalar Sample Sort Test: Many duplicates ($A[i]=i^{16}$ mod floor($\log_2 n$))

algo=ssssort
algo=stdsort

Run Time / $n \log_2 n$ [Nanoseconds]

Item Count [$\log_2(n)$]

Super Scalar Sample Sort Test: Few spikes, lots of noise

Super Scalar Sample Sort Test: All Ones