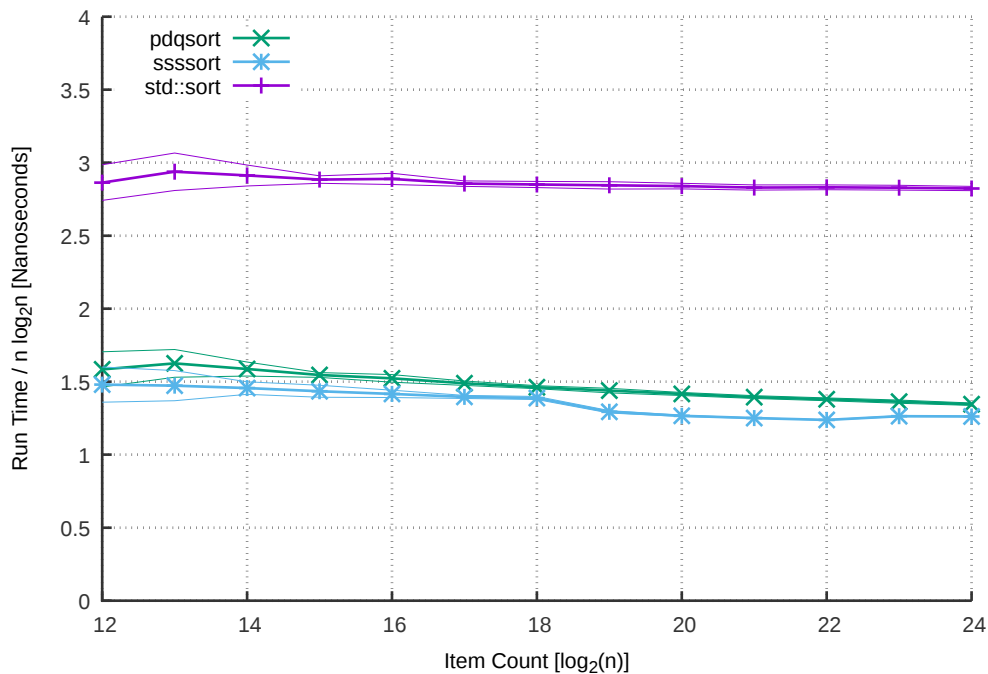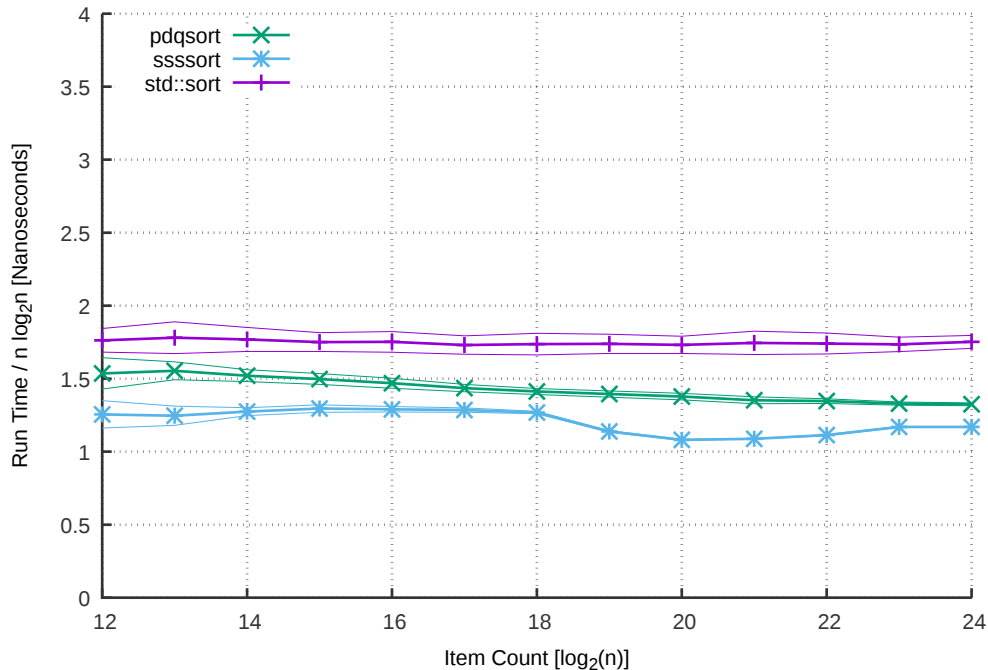Super Scalar Sample Sort Test: Random
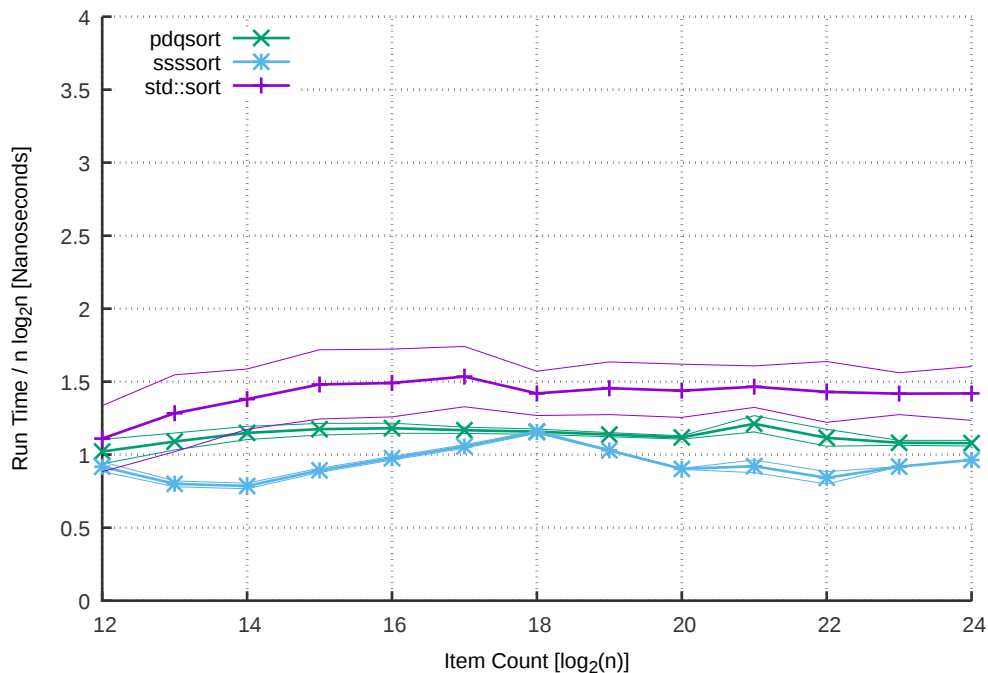
Super Scalar Sample Sort Test: 80% Sorted

Plot showing "Run Time / n log₂n [Nanoseconds]" on the y-axis (range 0 to 4) versus "Item Count [log₂(n)]" on the x-axis (range 12 to 24) for three sorting algorithms: pdqsort, ssssort, and std::sort.

Super Scalar Sample Sort Test: 90% Sorted

Legend:
- pdqsort
- ssssort
- std::sort

Y-axis: Run Time / n log₂n [Nanoseconds]
X-axis: Item Count [log₂(n)]

Super Scalar Sample Sort Test: 99% Sorted

Legend:
- pdqsort
- ssssort
- std::sort

Y-axis: Run Time / n log$_2$n [Nanoseconds]
X-axis: Item Count [log$_2$(n)]

Super Scalar Sample Sort Test: 99.9% Sorted

Run Time / n log₂n [Nanoseconds] (y-axis, labeled 0 to 4)

Item Count [log₂(n)] (x-axis, labeled 12 to 24)

Legend:
- pdqsort
- sssort
- std::sort
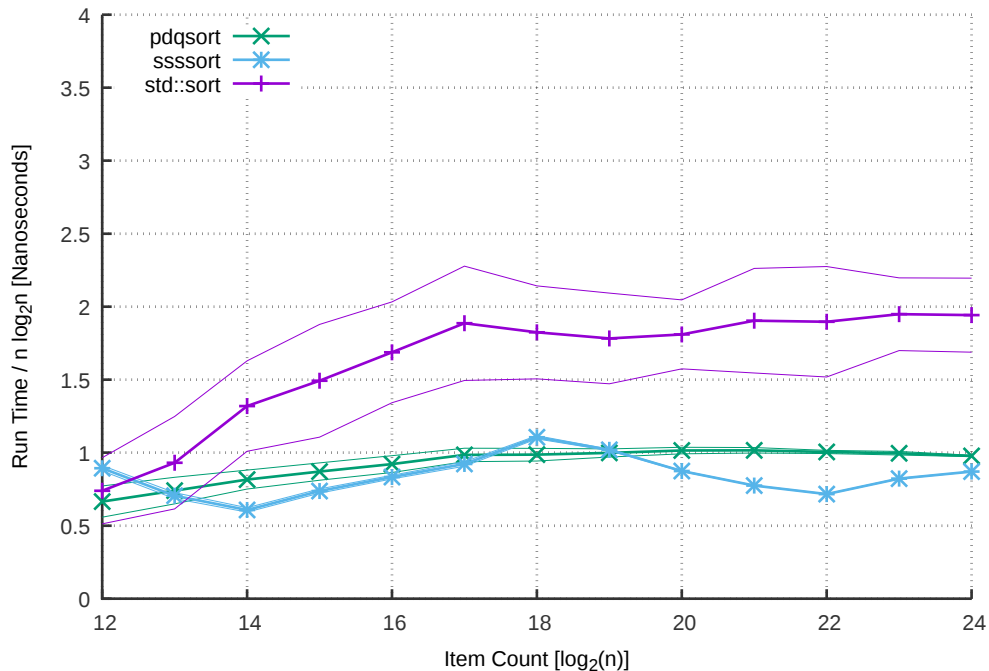
Super Scalar Sample Sort Test: 90% Sorted + 10% Random Tail

Legend:
- pdqsort
- sssssort
- std::sort

Y-axis: Run Time / n log$_2$n [Nanoseconds]

X-axis: Item Count [log$_2$(n)]
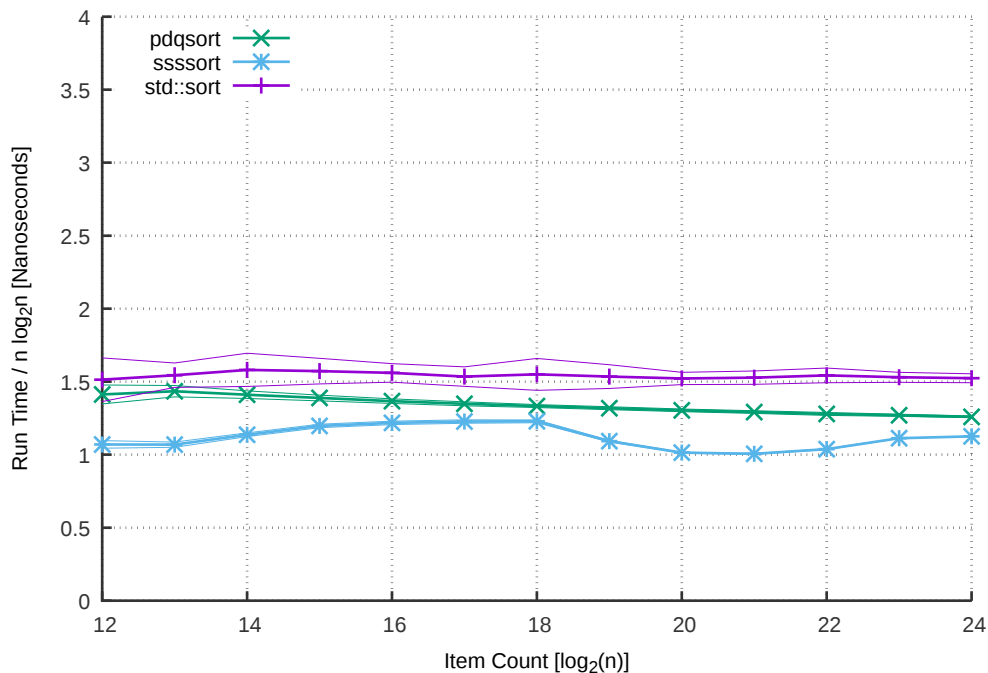
Super Scalar Sample Sort Test: 99% Sorted + 1% Random Tail
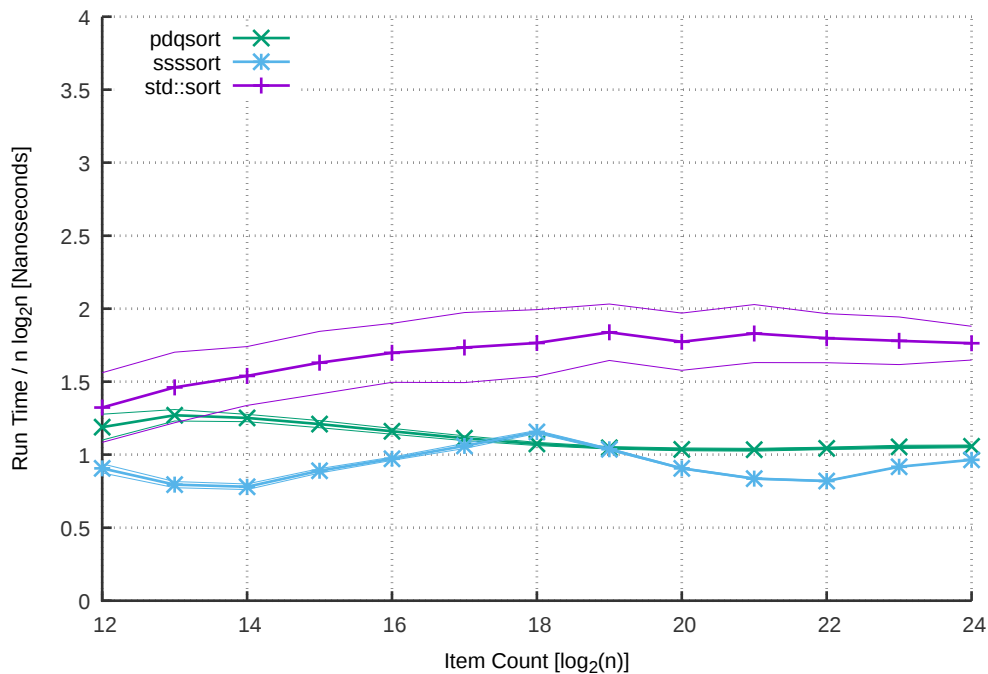
Super Scalar Sample Sort Test: Sorted

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

Item Count [log$_2$(n)]

pdqsort
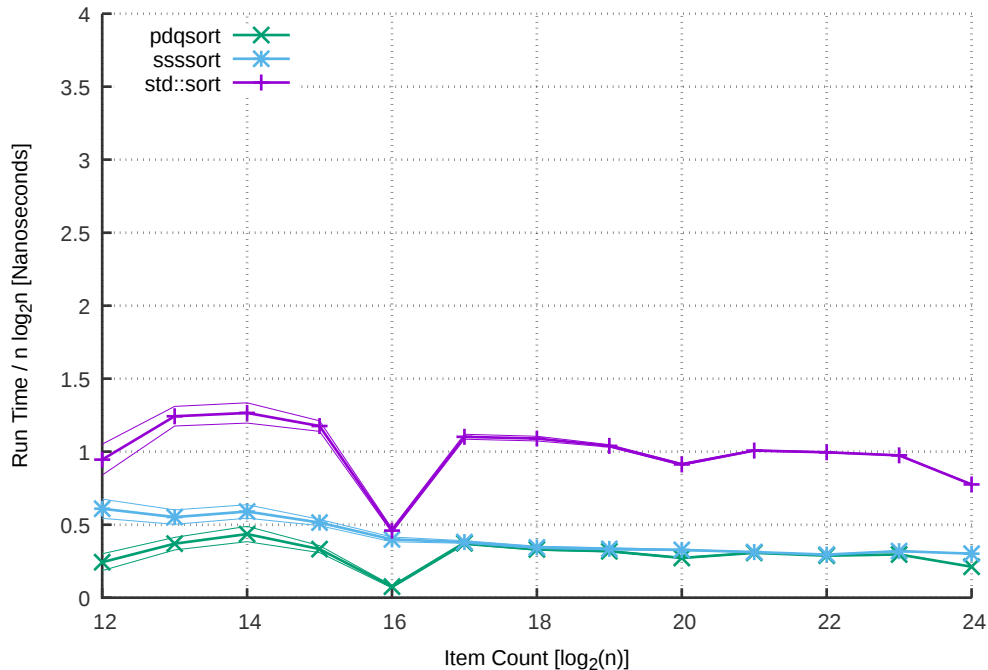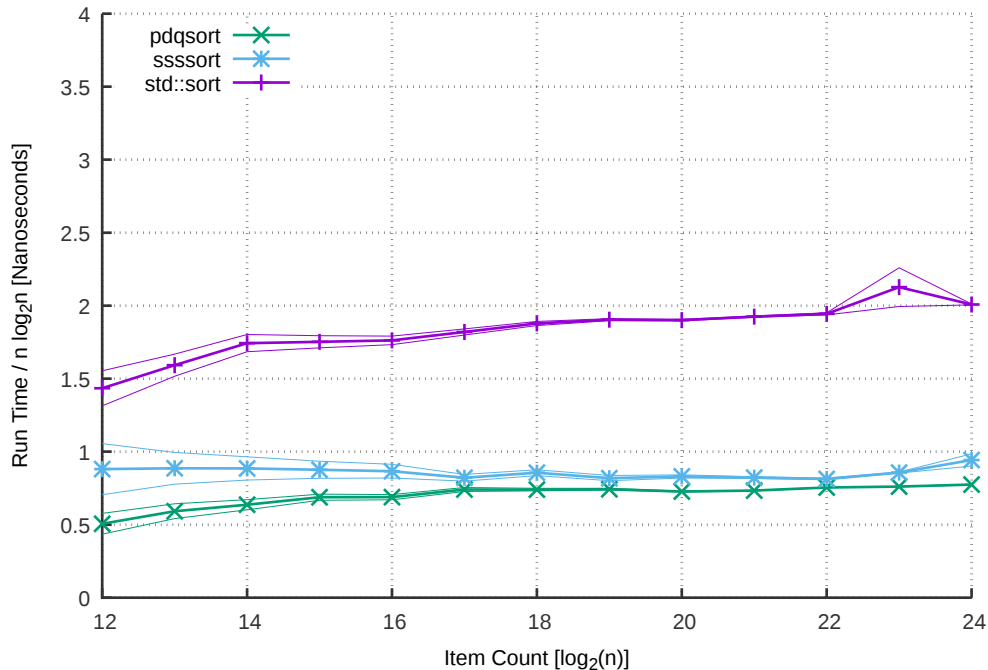ssssort
std::sort
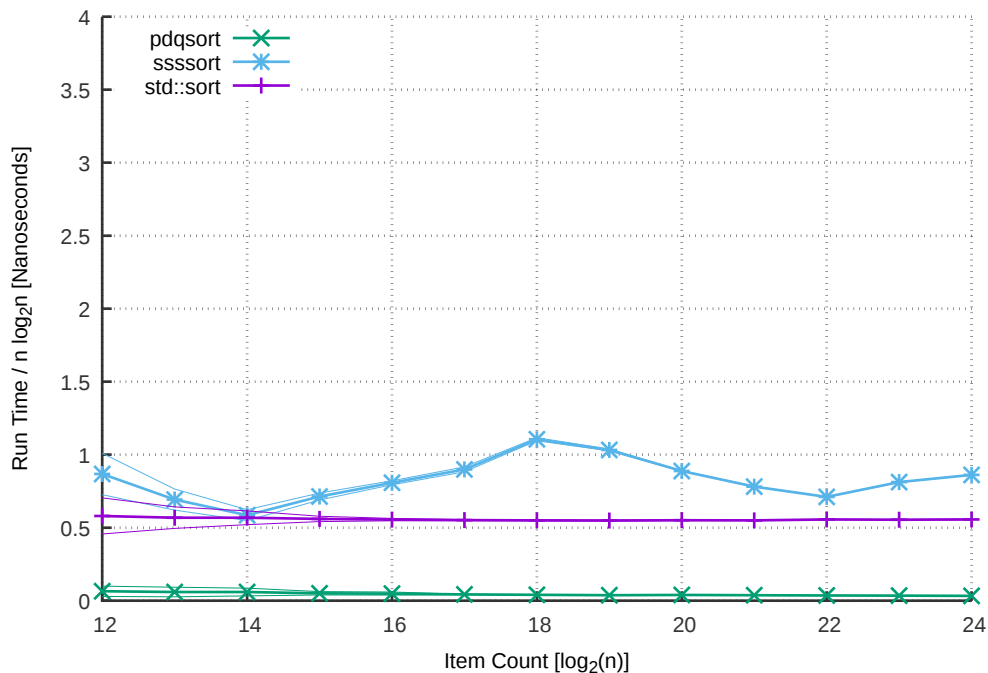
Super Scalar Sample Sort Test: Reverse Sorted
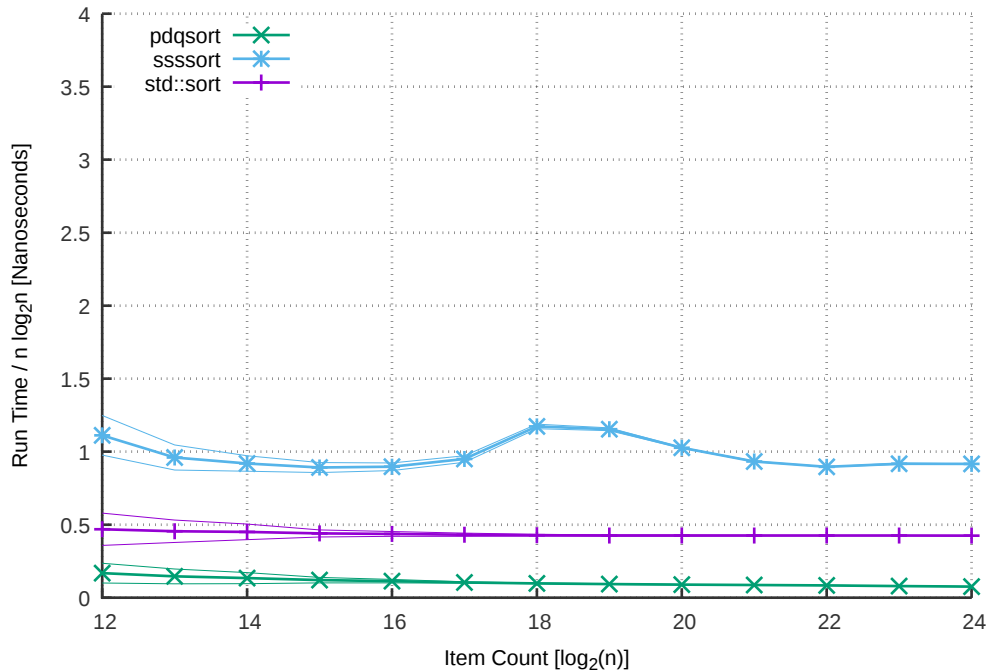
Super Scalar Sample Sort Test: Many duplicates ($A[i]=i^{16}$ mod floor($\log_2 n$))

Legend:
- pdqsort (green)
- sssssort (light blue)
- std::sort (magenta)

Y-axis: Run Time / n $\log_2$n [Nanoseconds]
X-axis: Item Count [$\log_2(n)$]

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

Super Scalar Sample Sort Test: All Ones

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

Item Count [$\log_2(n)$]

pdqsort
ssssort
std::sort