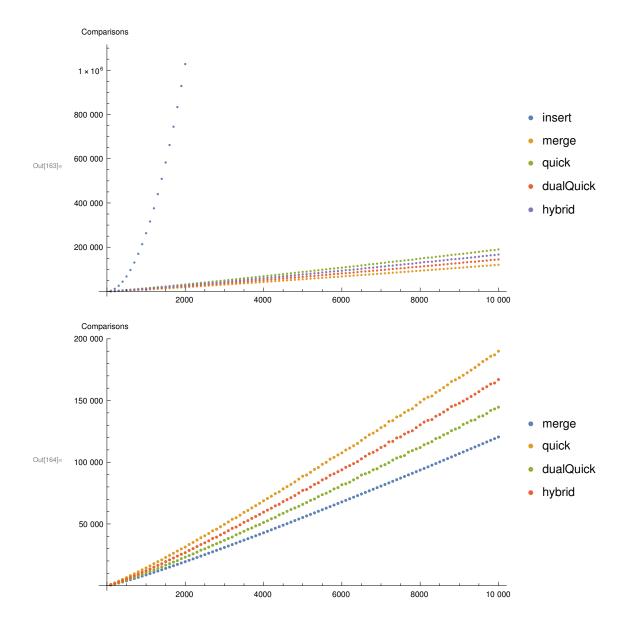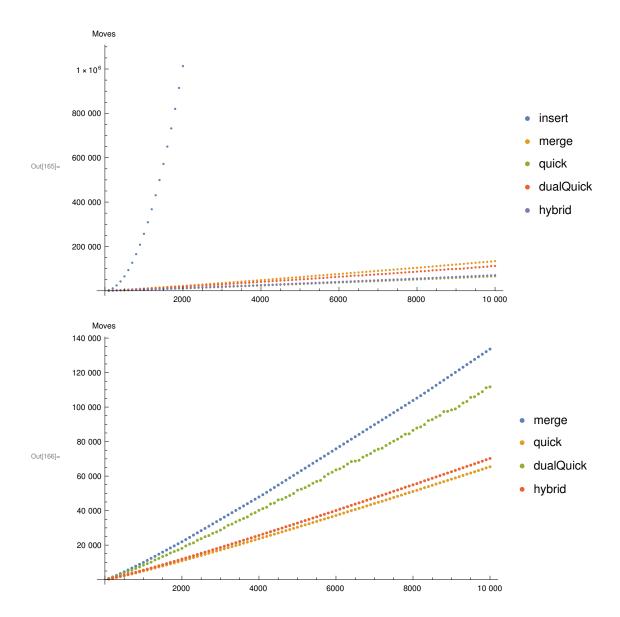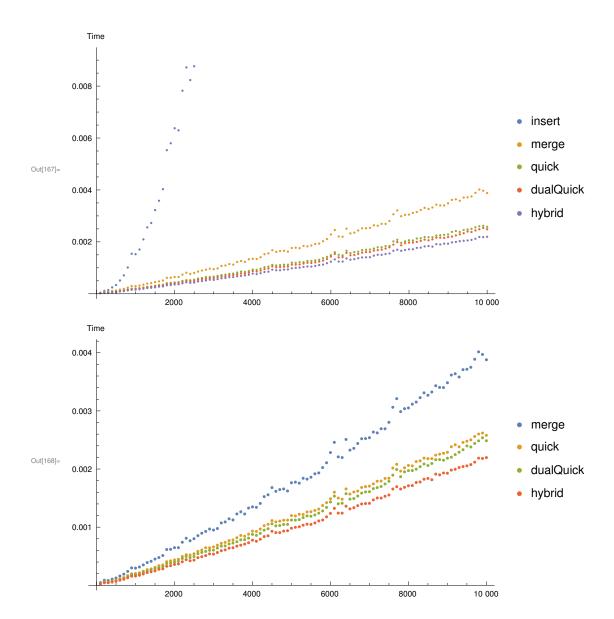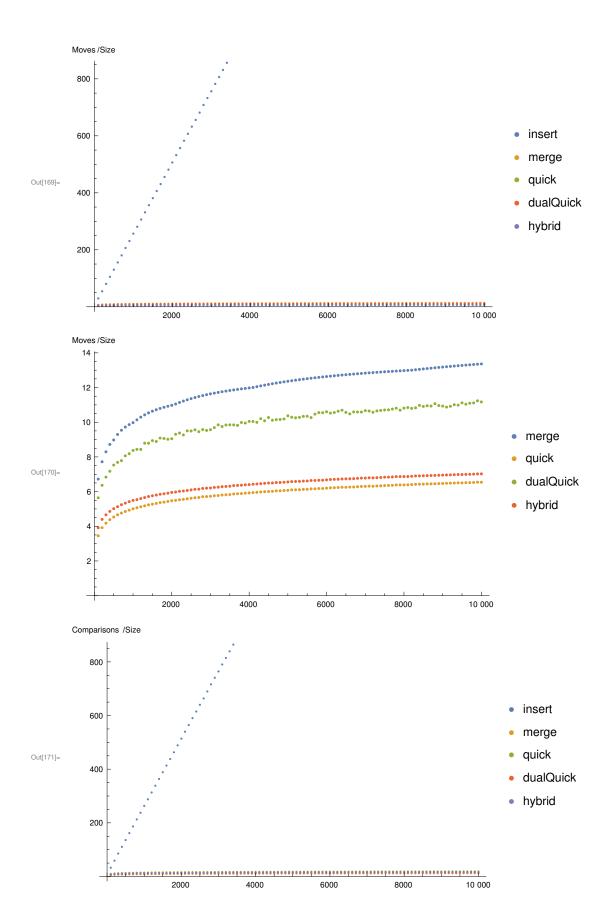```
ListPlot[{iComparisons , mComparisons , qComparisons , dComparisons , hComparisons },
  AxesLabel → "Comparisons ",
  PlotLegends → {"insert", "merge", "quick", "dualQuick", "hybrid"}]
ListPlot[{mComparisons , qComparisons , dComparisons , hComparisons },
  AxesLabel → "Comparisons ", PlotLegends → {"merge", "quick", "dualQuick", "hybrid"}]

ListPlot[{iMoves , mMoves , qMoves , dMoves , hMoves}, AxesLabel → "Moves" ,
  PlotLegends → {"insert", "merge", "quick", "dualQuick", "hybrid"}]
ListPlot[{mMoves , qMoves , dMoves , hMoves}, AxesLabel → "Moves",
  PlotLegends → {"merge", "quick", "dualQuick", "hybrid"}]

ListPlot[{iTime , mTime , qTime , dTime , hTime}, AxesLabel → "Time" ,
  PlotLegends → {"insert", "merge", "quick", "dualQuick", "hybrid"}]
ListPlot[{mTime , qTime , dTime , hTime}, AxesLabel → "Time",
  PlotLegends → {"merge", "quick", "dualQuick", "hybrid"}]

ListPlot[{iMovesToSize , mMovesToSize , qMovesToSize , dMovesToSize , hMovesToSize},
  AxesLabel → "Moves/Size" ,
  PlotLegends → {"insert", "merge", "quick", "dualQuick", "hybrid"}]
ListPlot[{mMovesToSize , qMovesToSize , dMovesToSize , hMovesToSize },
  AxesLabel → "Moves/Size", PlotLegends → {"merge", "quick", "dualQuick", "hybrid"}]

ListPlot[{iCompToSize , mCompToSize , qCompToSize , dCompToSize , hCompToSize },
  AxesLabel → "Comparisons /Size" ,
  PlotLegends → {"insert", "merge", "quick", "dualQuick", "hybrid"}]
ListPlot[{mCompToSize , qCompToSize , dCompToSize , hCompToSize},
  AxesLabel → "Comparisons /Size",
  PlotLegends → {"merge", "quick", "dualQuick", "hybrid"}]

f[x_] := Log[x];
With[{x := 100 xp}, tab = Table[{x, f[x]}, {xp, 1, 100}]];
ListPlot[{dCompToSize , tab},
  AxesLabel → "Comparisons /Size", PlotLegends → {"dualQuick", "log[n]"}]
ListPlot[dCompToSize / GroupBy[tab, First → Last, Mean],
  PlotLegends → {"Constant near n*lg(n) in dual-pivot Quicksort"}]
```

Comparisons

Out[163]=

- insert
- merge
- quick
- dualQuick
- hybrid

Comparisons

Out[164]=

- merge
- quick
- dualQuick
- hybrid

Moves

Out[165]=



- insert
- merge
- quick
- dualQuick
- hybrid

Moves

Out[166]=



- merge
- quick
- dualQuick
- hybrid

Out[167]=



Out[168]=

Moves /Size

Out[169]=



- insert
- merge
- quick
- dualQuick
- hybrid

Moves /Size

Out[170]=



- merge
- quick
- dualQuick
- hybrid

Comparisons /Size

Out[171]=



- insert
- merge
- quick
- dualQuick
- hybrid

Comparisons /Size

Out[172]=

- merge
- quick
- dualQuick
- hybrid

Comparisons /Size

Out[175]=

- dualQuick
- log[n]

Out[176]=

- Constant near n*lg(n) in dual-piv