```
> restart
```

Load Libraries

```
> with(plots)
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,          (1.1)
    conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d,
    densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d,
    implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot,
    listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot,
    matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot,
    polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus,
    semilogplot, setcolors, setoptions, setoptions3d, shadebetween, spacecurve,
    sparsematrixplot, surfdata, textplot, textplot3d, tubeplot]
> with(LinearAlgebra)
[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm,      (1.2)
    BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column,
    ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix,
    CompressedSparseForm, ConditionNumber, ConstantMatrix, ConstantVector, Copy,
    CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant,
    Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct,
    EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute,
    FrobeniusForm, FromCompressedSparseForm, FromSplitForm, GaussianElimination,
    GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape,
    GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm,
    HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix,
    IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary,
    JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUDecomposition,
    LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential,
    MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower,
    MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular,
    Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent,
    Pivot, PopovForm, ProjectionMatrix, QRDecomposition, RandomMatrix,
    RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row,
    RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector,
    SchurForm, SingularValues, SmithForm, SplitForm, StronglyConnectedBlocks,
    SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace,
    Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd,
    VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix,
    ZeroVector, Zip]
> with(Statistics)
[AbsoluteDeviation, AgglomeratedPlot, AreaChart, AutoCorrelation, AutoCorrelationPlot,      (1.3)
    BarChart, Biplot, Bootstrap, BoxPlot, BubblePlot, CDF, CGF, CentralMoment,
CharacteristicFunction, ChiSquareGoodnessOfFitTest, ChiSquareIndependenceTest,
ChiSquareSuitableModelTest, ColumnGraph, Correlation, CorrelationMatrix,
Correlogram, Count, CountMissing, Covariance, CovarianceMatrix, CrossCorrelation,
Cumulant, CumulantGeneratingFunction, CumulativeDistributionFunction,
CumulativeProduct, CumulativeSum, CumulativeSumChart, DataSummary, Decile,
DensityPlot, Detrend, Difference, DiscreteValueMap, Distribution, ErrorPlot,
EvaluateToFloat, Excise, ExpectedValue, ExponentialFit, ExponentialSmoothing,
FailureRate, FisherInformation, Fit, FivePointSummary, FrequencyPlot,
FrequencyTable, GeometricMean, GridPlot, HarmonicMean, HazardRate, HeatMap,
Histogram, HodgesLehmann, Information, InteractiveDataAnalysis, InterquartileRange,
InverseSurvivalFunction, Join, KernelDensity, KernelDensityPlot,
KernelDensitySample, Kurtosis, LeastTrimmedSquares, Likelihood,
LikelihoodRatioStatistic, LineChart, LinearFilter, LinearFit, LogLikelihood,
LogarithmicFit, Lowess, MGF, MLE, MakeProcedure, MaximumLikelihoodEstimate,
Mean, MeanDeviation, Median, MedianDeviation, MillsRatio, Mode, Moment,
MomentGeneratingFunction, MovingAverage, MovingMedian, MovingStatistic,
NonlinearFit, NormalPlot, OneSampleChiSquareTest, OneSampleTTest,
OneSampleZTest, OneWayANOVA, OrderByRank, OrderStatistic, PCA, PDF,
ParetoChart, Percentile, PieChart, PointPlot, PolynomialFit, PowerFit,
PredictiveLeastSquares, PrincipalComponentAnalysis, Probability,
ProbabilityDensityFunction, ProbabilityFunction, ProbabilityPlot, ProfileLikelihood,
ProfileLogLikelihood, QuadraticMean, Quantile, QuantilePlot, Quartile,
RandomVariable, Range, Rank, Remove, RemoveInRange, RemoveNonNumeric,
RepeatedMedianEstimator, RousseeuwCrouxQn, RousseeuwCrouxSn, Sample, Scale,
ScatterPlot, ScatterPlot3D, Score, ScreePlot, Select, SelectInRange, SelectNonNumeric,
ShapiroWilkWTest, Shuffle, Skewness, Sort, Specialize, SplitByColumn,
StandardDeviation, StandardError, StandardizedMoment, SunflowerPlot, Support,
SurfacePlot, SurvivalFunction, SymmetryPlot, Tally, TallyInto, TreeMap, Trim,
TrimmedMean, TwoSampleFTest, TwoSamplePairedTTest, TwoSampleTTest,
TwoSampleZTest, Variance, Variation, VennDiagram, ViolinPlot, WeibullPlot,
WeightedMovingAverage, Winsorize, WinsorizedMean]
> currentdir( )
"F:\Users\Kenne.DESKTOP-BT6VROU\Documents\GitHub\CudaLearn\Reduciton\CpuSingle"      (1)
> FileL := FileTools[ListDirectory](".", 'all', 'returnonly' = "·.txt")
                              FileL := ["Speed.txt"]                                 (2)
> Type = "i5−2500 K CPU @ 3.30 GHz"
                         Type = "i5-2500 K CPU @ 3.30 GHz"                           (3)
> FamSpeed := ImportMatrix(FileL[1], source = csv)
```

$$FamSpeed := \left[\begin{array}{ccc} 100. & 1.30000000000000\ 10^{-6} & 2.66366000000000\ 10^{-8} \\ 133. & 1.50000000000000\ 10^{-6} & 1.50143000000000\ 10^{-7} \\ 177. & 9.00000000000000\ 10^{-7} & -3.37294000000000\ 10^{-7} \\ 236. & 3.00000000000000\ 10^{-6} & 2.98573000000000\ 10^{-7} \\ 315. & 1.40000000000000\ 10^{-6} & 2.29747000000000\ 10^{-7} \\ 420. & 5.20000000000000\ 10^{-6} & 1.55147000000000\ 10^{-7} \\ 560. & 1.90000000000000\ 10^{-6} & -1.94351000000000\ 10^{-8} \\ 747. & 9.00000000000000\ 10^{-6} & -3.93865000000000\ 10^{-7} \\ 996. & 3.20000000000000\ 10^{-6} & -8.63760000000000\ 10^{-9} \\ 1328. & 4.10000000000000\ 10^{-6} & -2.43901000000000\ 10^{-7} \\ \vdots & \vdots & \vdots \end{array}\right] \tag{4}$$

54 × 3 Matrix

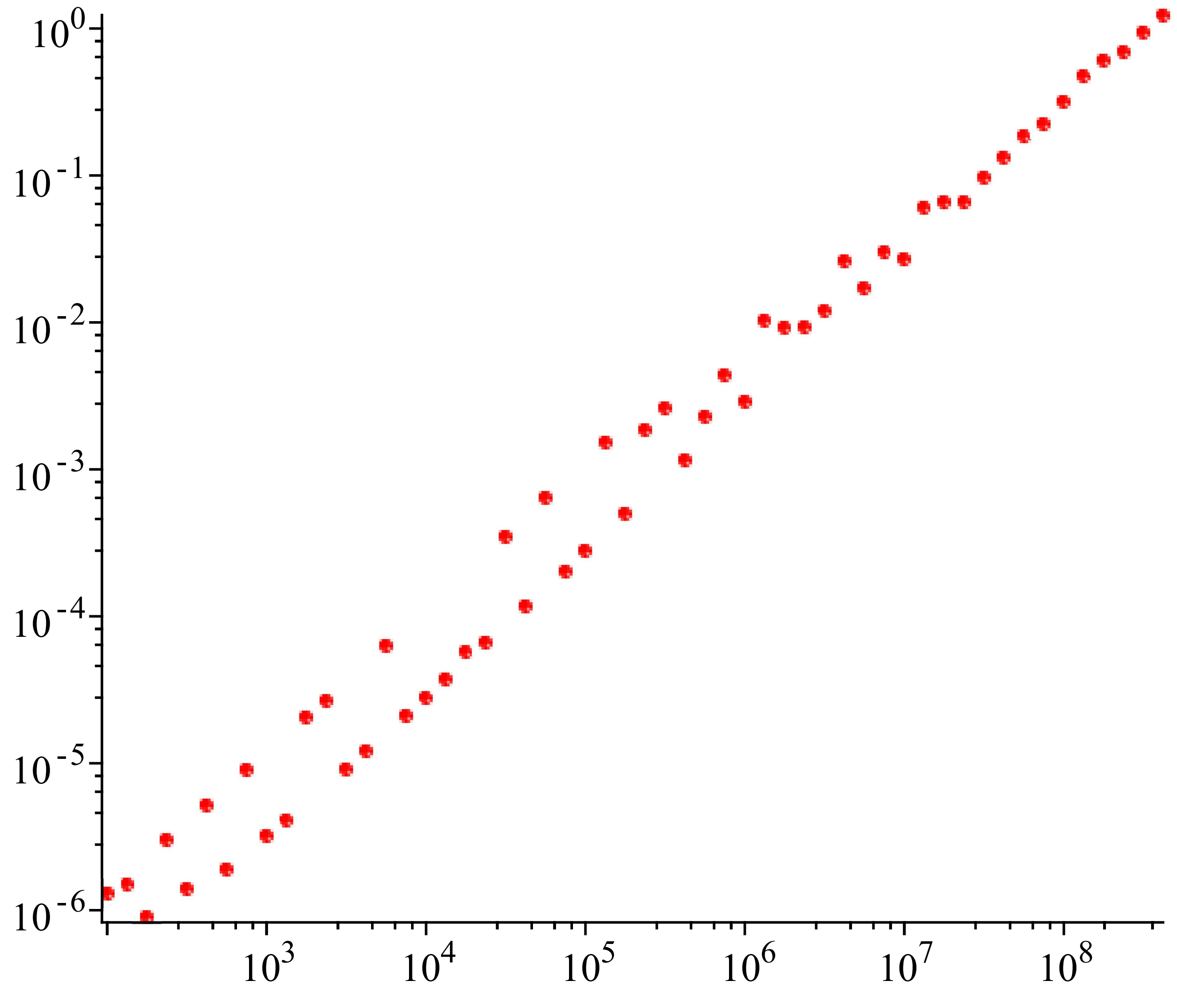
> *N* := *Transpose*(*FamSpeed*)[1];

$$N := [100., 133., 177., 236., 315., 420., 560., 747., 996., 1328., 1771., 2362., 3150., 4201., 5602., 7470., 9961., 13283., 17713., 23621., 31499., 42005., 56015., 74697., 99610., 132832., 177134., 236212., 314994., 420051., 560147., 746968., 996098., 1.328318\ 10^{6}, 1.771341\ 10^{6}, 2.362121\ 10^{6}, 3.149939\ 10^{6}, 4.200511\ 10^{6}, 5.601471\ 10^{6}, 7.469682\ 10^{6}, 9.960981\ 10^{6}, 1.3283182\ 10^{7}, 1.7713408\ 10^{7}, 2.3621209\ 10^{7}, 3.1499388\ 10^{7}, 4.2005109\ 10^{7}, 5.6014713\ 10^{7}, 7.4696820\ 10^{7}, 9.9609810\ 10^{7}, 1.32831816\ 10^{8}, 1.77134074\ 10^{8}, 2.36212084\ 10^{8}, 3.14993877\ 10^{8}, 4.20051086\ 10^{8}] \tag{5}$$

> *DeltaT* := *Transpose*(*FamSpeed*)[2];

$$DeltaT := [1.30000000000000\ 10^{-6}, 1.50000000000000\ 10^{-6}, 9.00000000000000\ 10^{-7}, 3.00000000000000\ 10^{-6}, 1.40000000000000\ 10^{-6}, 5.20000000000000\ 10^{-6}, 1.90000000000000\ 10^{-6}, 9.00000000000000\ 10^{-6}, 3.20000000000000\ 10^{-6}, 4.10000000000000\ 10^{-6}, 0.0000206000000000000, 0.0000266000000000000, 9.10000000000000\ 10^{-6}, 0.0000121000000000000, 0.0000631000000000000, 0.0000210000000000000, 0.0000279000000000000, 0.0000372000000000000, 0.0000572000000000000, 0.0000664000000000000, 0.000348900000000000, 0.000116800000000000, 0.000643800000000000, 0.000201500000000000, 0.000279700000000000, 0.00152780000000000, 0.000497700000000000, 0.00185010000000000, 0.00260760000000000, 0.00115830000000000, 0.00228620000000000, 0.00436600000000000, 0.00287780000000000, 0.0102924000000000, 0.00919570000000000, 0.00926450000000000, 0.0119864000000000, 0.0260595000000000, 0.0171510000000000, 0.0302298000000000, 0.0270736000000000, 0.0605652000000000, 0.0659323000000000, 0.0659654000000000, 0.0965747000000000, \tag{6}$$

```
    0.132370000000000, 0.185577000000000, 0.224490000000000, 0.317585000000000,
    0.473895000000000, 0.604814000000000, 0.694554000000000, 0.937484000000000,
    1.23193000000000 ]
> Err := Transpose(FamSpeed)[3];
Err := [2.66366000000000 10^-8, 1.50143000000000 10^-7, -3.37294000000000 10^-7,        (7)
    2.98573000000000 10^-7, 2.29747000000000 10^-7, 1.55147000000000 10^-7,
    -1.94351000000000 10^-8, -3.93865000000000 10^-7, -8.63760000000000 10^-9,
    -2.43901000000000 10^-7, 6.02531000000000 10^-7, -1.49169000000000 10^-6,
    -8.96918000000000 10^-7, 1.46654000000000 10^-7, 9.37750000000000 10^-7,
    2.23334000000000 10^-6, -2.43484000000000 10^-6, -2.53902000000000 10^-6,
    -2.25544000000000 10^-6, 1.36195000000000 10^-6, -9.30775000000000 10^-7,
    6.20667000000000 10^-6, 8.43616000000000 10^-7, 3.99471000000000 10^-6,
    -2.92095000000000 10^-6, 3.42400000000000 10^-6, 4.74537000000000 10^-7,
    -0.0000144450000000000, -3.94483000000000 10^-6, -3.72046000000000 10^-6,
    5.74396000000000 10^-6, 5.11234000000000 10^-7, -3.26537000000000 10^-6,
    -2.45651000000000 10^-6, -8.69195000000000 10^-6, 9.95978000000000 10^-6,
    6.64839000000000 10^-6, -9.54467000000000 10^-6, -2.82228000000000 10^-6,
    0.0000443274000000000, 0.0000579896000000000, 0.0000355613000000000,
    0.0000220012000000000, -0.0000895167000000000, 0.0000755606000000000,
    -0.201185000000000, -0.400953000000000, -0.550778000000000,
    -0.663144000000000, -0.747364000000000, -0.810562000000000,
    -0.857948000000000, -0.893476000000000, -0.920117000000000 ]
> PlotOpts0 := style=point, symbol=solidcircle, color=red, symbolsize=8
        PlotOpts0 := style=point, symbol=solidcircle, color=red, symbolsize=8            (8)
> PData := loglogplot(N, DeltaT, PlotOpts0)
```



```
> LogN := map(log, N);
LogN := [4.60517018598809, 4.89034912822175, 5.17614973257383, 5.46383180502561,      (9)
    5.75257263882563, 6.04025471127741, 6.32793678372919, 6.61606518513282,
    6.90374725758460, 7.19142933003638, 7.47929963778283, 7.76726399675731,
    8.05515773181968, 8.34307787116938, 8.63087895582005, 8.91865027812686,
    9.20643274714516, 9.49424030113250, 9.78205411225260, 10.0698914258577,
    10.3577110782781, 10.6455439377989, 10.9333747909928, 11.2211952096719,
    11.5090178401392, 11.7968404508528, 12.0846617872759, 12.3724849858579,
    12.6603088700073, 12.9481314114591, 13.2359545282642, 13.5237776251786,
    13.8116009253007, 14.0994240381844, 14.3872474448141, 14.6750705022004,
    14.9628936455349, 15.2507167425195, 15.5385397997828, 15.8263629859217,
    16.1141861186865, 16.4020092817448, 16.6898324248993, 16.9776555528596,
    17.2654786750357, 17.5533018117069, 17.8411249463342, 18.1289480789158,
    18.4167712116805, 18.7045943446256, 18.9924174840207, 19.2802406204528,
    19.5680637585057, 19.8558868951783]
> LogDeltaT := map(log, DeltaT);
LogDeltaT := [-13.5531462934968, -13.4100454498561, -13.9208710736221,      (10)
    -12.7168982692962, -13.4790383213431, -12.1668519323769,
−13.1736566717919, −11.6182859806281, −12.6523597481586,
−12.4045235842540, −10.7902194821687, −10.5345993421766,
−11.6072361444415, −11.3223051053616, −9.67078978841711,
−10.7709881202409, −10.4868838691370, −10.1992017966852,
−9.76895665957852, −9.61981350148189, −7.96072520977155,
−9.05504748757015, −7.34812243909957, −8.50972117657754,
−8.18179295775636, −6.48392648685701, −7.60551307209977,
−6.29251558729872, −5.94932502091389, −6.76080186602597,
−6.08086422837260, −5.43390802085438, −5.85072916563496,
−4.57634952017990, −4.68901929549285, −4.68156538724180,
−4.42398260523534, −3.64737289396221, −4.06569879803219,
−3.49892708615708, −3.60919619564991, −2.80403480828156,
−2.71912682098081, −2.71862491684306, −2.33743847682861,
−2.02215424725646, −1.68428538872285, −1.49392411622176,
−1.14700977994363, −0.746769500813357, −0.502834306236580,
−0.364485366030709, −0.0645555879498751, 0.208582045315296 ]
```

> $LogLogFit := Fit(a \cdot x + b, LogN, LogDeltaT, x)$

$$LogLogFit := 0.943613956315727\, x - 18.5199994827706 \quad \textbf{(11)}$$

> $LinFit := simplify(\exp(subs(x = \log(n), LogLogFit)))$

$$LinFit := 9.05454058667457\ 10^{-9}\ n^{0.9436139563} \quad \textbf{(12)}$$

> $PlotOpts1 := color = blue, thickness = 3$

$$PlotOpts1 := color = blue, thickness = 3 \quad \textbf{(13)}$$

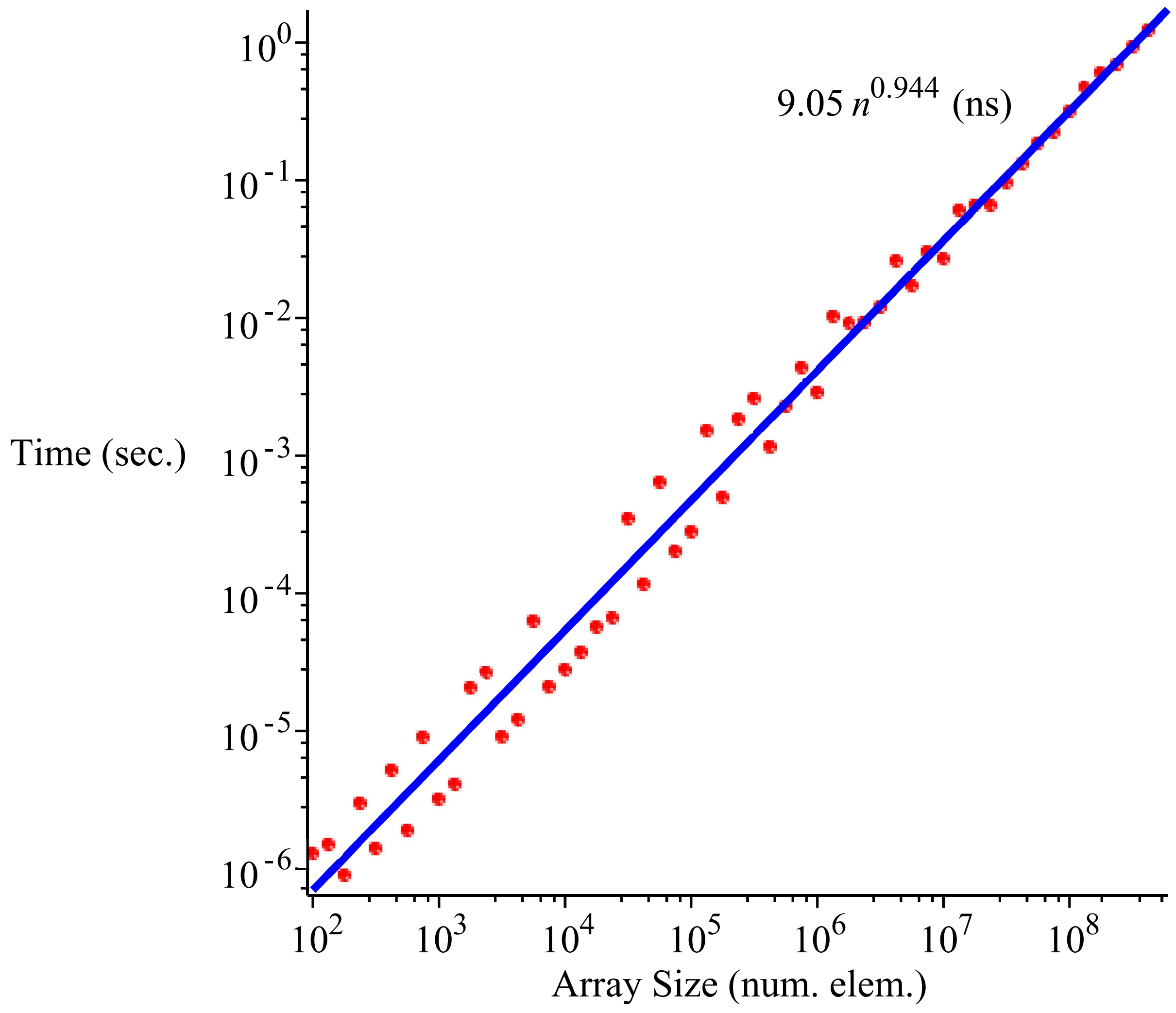
> $PFit := loglogplot(LinFit, n = 1e2 .. 6e8, PlotOpts1)$:

> $PLegend := textplot([4e6, 0.4, typeset(9.05 \cdot n^{0.944}, \text{" (ns)"})])$:

> $PlotOpts2 := labels = [$"Array Size (num. elem.)", "Time (sec.)"$]$;

$$PlotOpts2 := labels = [\text{"Array Size (num. elem.)", "Time (sec.)"}] \quad \textbf{(14)}$$

> $display(PData, PFit, PLegend, PlotOpts2)$



Not a good fit

```
> LogLogFitA := Fit(x + b, LogN, LogDeltaT, x)
```
$$LogLogFitA := -19.2095233430932 + x \tag{2.1}$$

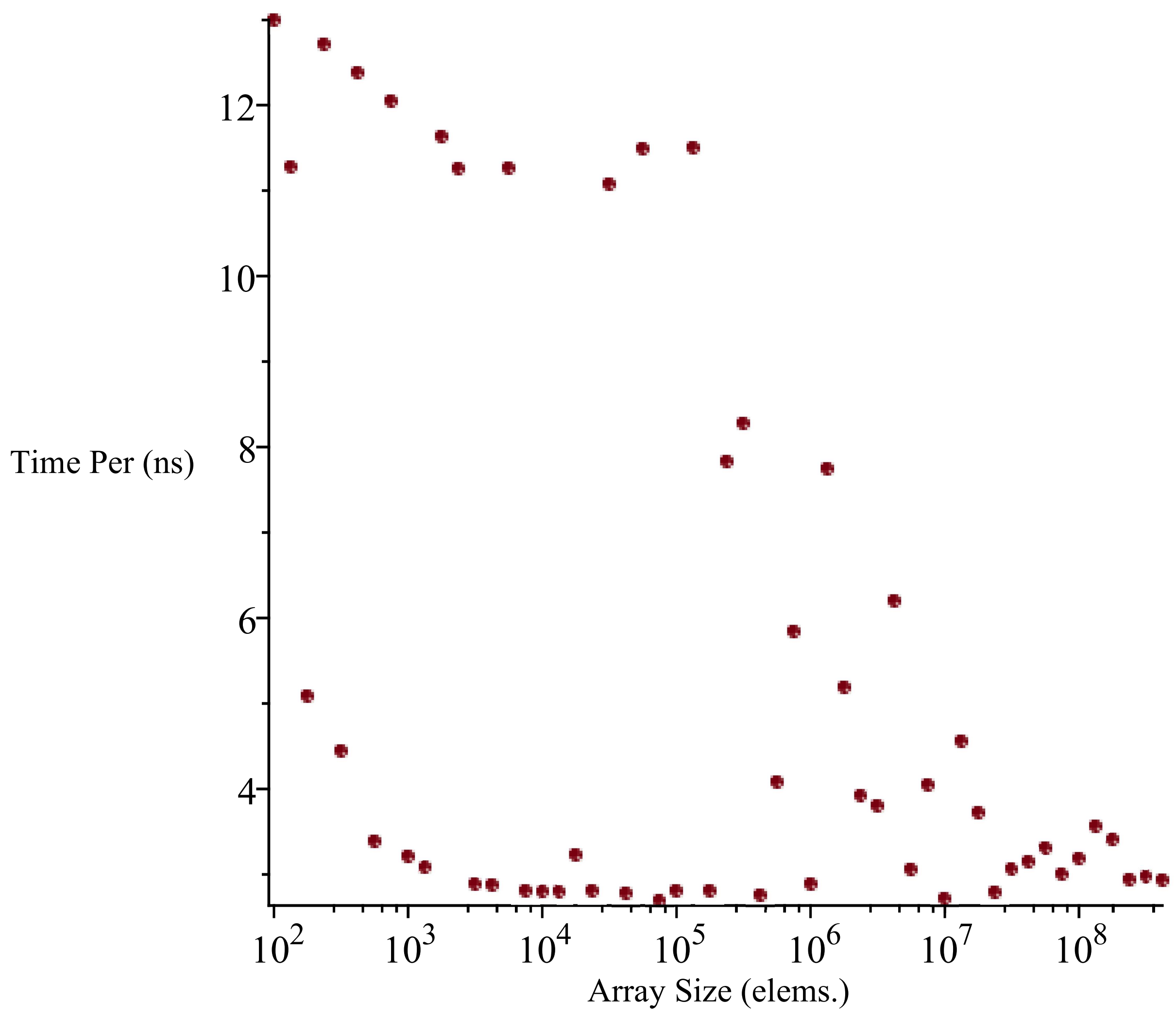
```
> LinFitA := simplify(exp(subs(x = ln(n), LogLogFitA)))
```
$$LinFitA := 4.54370379735632\ 10^{-9}\ n \tag{2.2}$$

```
> PLegendA := textplot([4e6, 0.4, typeset(4.54· n, " (ns)")]) :
> display(PData, PFitA, PLegendA, PlotOpts2)
Error, (in plots:-display) expecting plot structure but received: PFitA
> NumPoint := numelems(N)
```
$$NumPoint := 54 \tag{15}$$

$$> TimePer := \left[seq\left(\frac{DeltaT[k]}{N[k]}, k = 1 ..NumPoint\right)\right];$$

$$TimePer := [1.30000000000000\ 10^{-8}, 1.12781954887218\ 10^{-8}, 5.08474576271186\ 10^{-9}, 1.27118644067797\ 10^{-8}, 4.44444444444444\ 10^{-9}, 1.23809523809524\ 10^{-8}, 3.39285714285714\ 10^{-9}, 1.20481927710843\ 10^{-8}, 3.21285140562249\ 10^{-9}, 3.08734939759036\ 10^{-9}, 1.16318464144551\ 10^{-8}, 1.12616426756986\ 10^{-8}, \tag{16}$$

```
2.88888888888889 10^-9, 2.88026660318972 10^-9, 1.12638343448768 10^-8,
2.81124497991968 10^-9, 2.80092360204799 10^-9, 2.80057215990364 10^-9,
3.22926664032067 10^-9, 2.81105795690276 10^-9, 1.10765421124480 10^-8,
2.78062135460064 10^-9, 1.14933499955369 10^-8, 2.69756482857411 10^-9,
2.80795100893485 10^-9, 1.15017465670923 10^-8, 2.80973726105660 10^-9,
7.83237092103703 10^-9, 8.27825291910322 10^-9, 2.75752230086347 10^-9,
4.08142862498594 10^-9, 5.84496256867764 10^-9, 2.88907316348391 10^-9,
7.74844577879694 10^-9, 5.19137760600585 10^-9, 3.92211067934285 10^-9,
3.80528003875631 10^-9, 6.20388805076335 10^-9, 3.06187428266611 10^-9,
4.04699959114725 10^-9, 2.71796522852518 10^-9, 4.55954002587633 10^-9,
3.72216910489500 10^-9, 2.79263436515887 10^-9, 3.06592305856863 10^-9,
3.15128333555806 10^-9, 3.31300456721076 10^-9, 3.00534882207837 10^-9,
3.18829039027381 10^-9, 3.56763171859368 10^-9, 3.41444187638342 10^-9,
2.94038301613731 10^-9, 2.97619753415080 10^-9, 2.93280993921773 10^-9]
> min(TimePer)
                        2.69756482857411 10^-9                        (17)
> PlotOptsB0 := style=point, symbol=solidcircle, symbolsize=8, labels = ["Array Size (elems.)",
      "Time Per (ns)"]
PlotOptsB := style=point, symbol=solidcircle, symbolsize=8, labels = ["Array Size (elems.)",   (18)
    "Time Per (ns)"]
> PTimePerData := semilogplot(N, TimePer·1e9, PlotOptsB0)
```



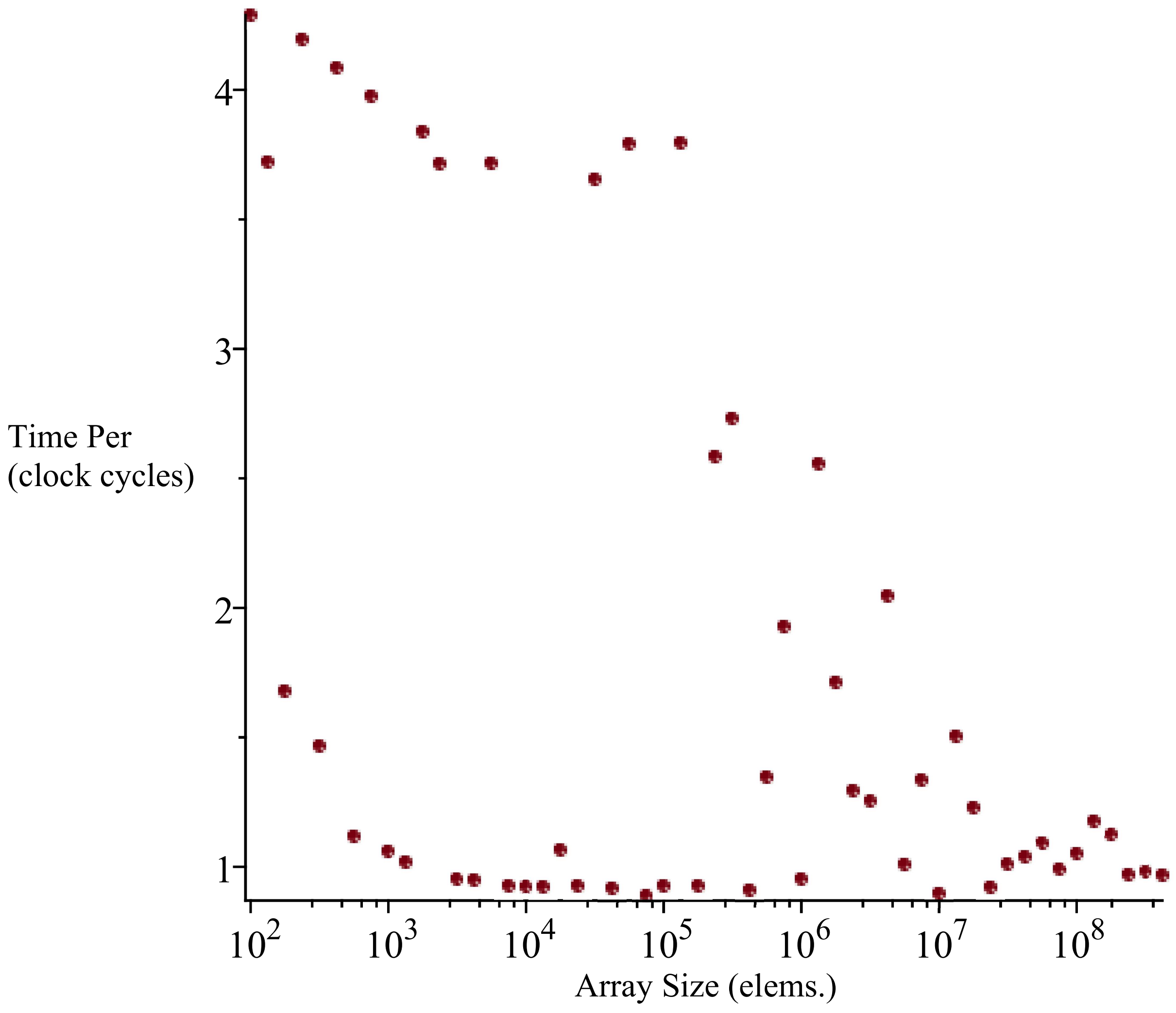
```
> PlotOptsB1 := style = point, symbol = solidcircle, symbolsize = 8, labels = ["Array Size (elems.)",
          "Time Per\n(clock cycles)"]
```

PlotOptsB1 := style = point, symbol = solidcircle, symbolsize = 8, labels = ["Array Size (elems.)", "Time Per
(clock cycles)"] **(19)**

```
> ClockRate := 3.3e8
```

$$ClockRate := 3.3\ 10^8 \quad \textbf{(20)}$$

```
> PTimePerData := semilogplot(N, TimePer·ClockRate, PlotOptsB1)
```



Time Per
(clock cycles)
4
3
2
1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
Array Size (elems.)

>