COMP114
Experimental Methods in Computing

Analysis and Presentation

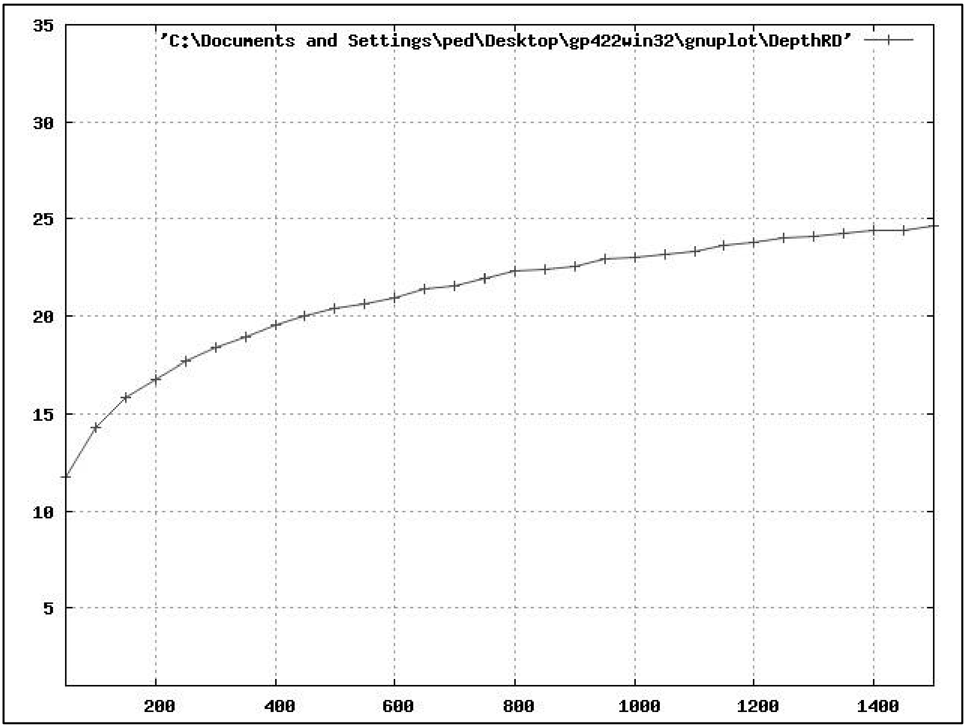
Background

- The three assessment tasks set so far have asked for the outcome of each experiment to be presented in the form of *tables*.
- This is often not the clearest method of detecting or arguing for a particular behaviour being present.

Example

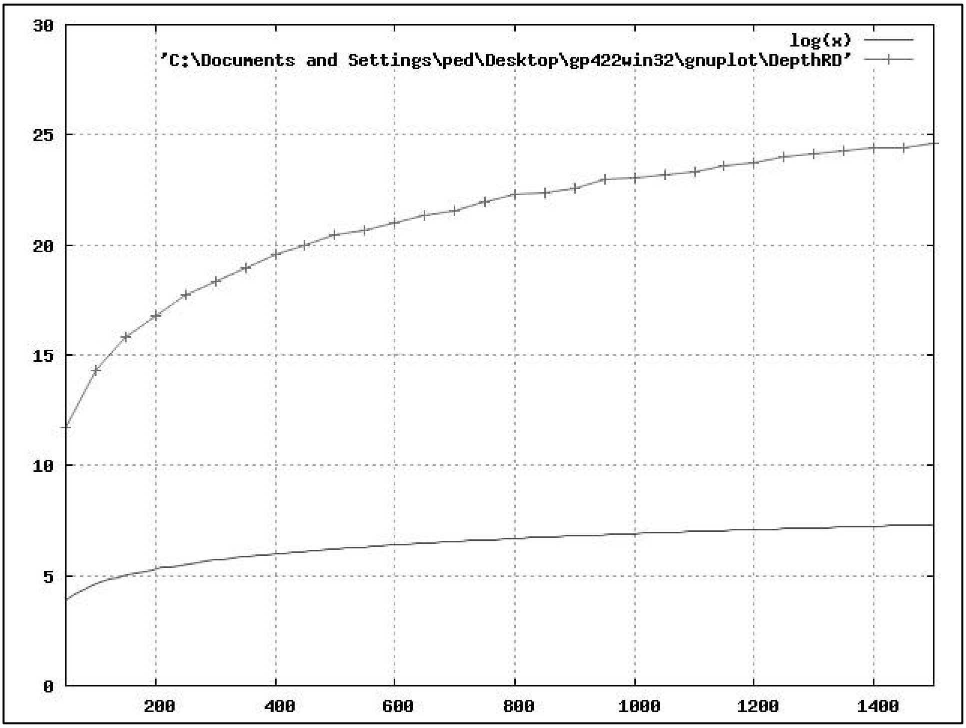
- A table of average depth for the RootDown random binary tree generator for n between 50 and 1500 in steps of 50 produced

50	11.278	800	22.32
100	14.285	850	22.383
150	15.813	900	22.591
200	16.793	950	22.95
250	17.726	1000	23.016
300	18.355	1050	23.185
350	18.95	1100	23.338
400	19.553	1150	23.616
450	19.983	1200	23.76
500	20.431	1250	24.013
550	20.626	1300	24.13
600	20.973	1350	24.243
650	21.373	1400	24.438
700	21.541	1450	24.43
750	21.97	1500	24.611



Advantages

- The “curve” indicates the rate at which the depth of random trees increases.
- The shape of such curves often allows hypotheses to be formed about how a measure behaves as size increases.
- For example on the preceding slide,



Similarity

- The “shape” of the data curve and the log(n) curve are “similar”.
- How can we use these to find a “better” match?
- Suppose, instead of listing the depth against n, we output depth/log(n).
- This results in the following table,

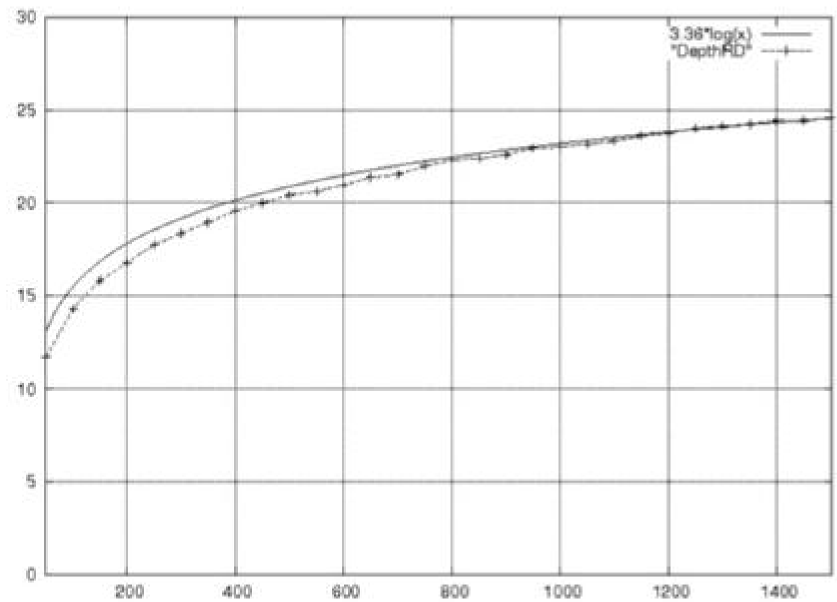
50	2.997937
100	3.101948
150	3.155887
200	3.169497
250	3.210382
300	3.21804
350	3.234929
400	3.263476
450	3.270943
500	3.287577
550	3.268822
600	3.278604
650	3.299844
700	3.28816
750	3.318694
800	3.339012

850	3.318342
900	3.321036
950	3.347207
1000	3.331907
1050	3.332832
1100	3.332541
1150	3.350967
1200	3.351163
1250	3.367458
1300	3.365355
1350	3.363412
1400	3.373445
1450	3.356083
1500	3.365275

Properties

- Examining the results it can be seen that the value depth/log(n) "appears" to *converge* to about 3.36.
- This motivates the following hypothesis:

  "*The average depth of n-leaf binary trees randomly generated by the RootDown method is* ≈ 3.36 log(n)"

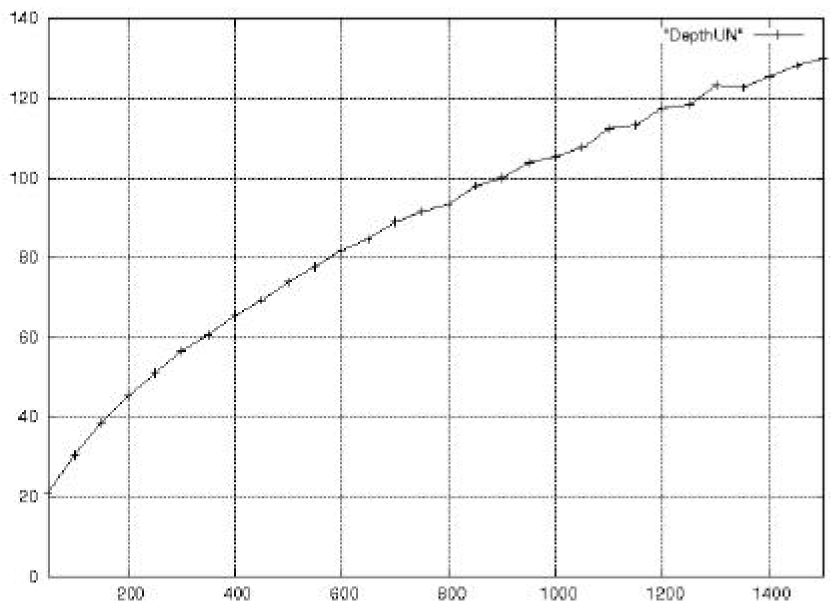


Depth in Uniformly Generated Trees

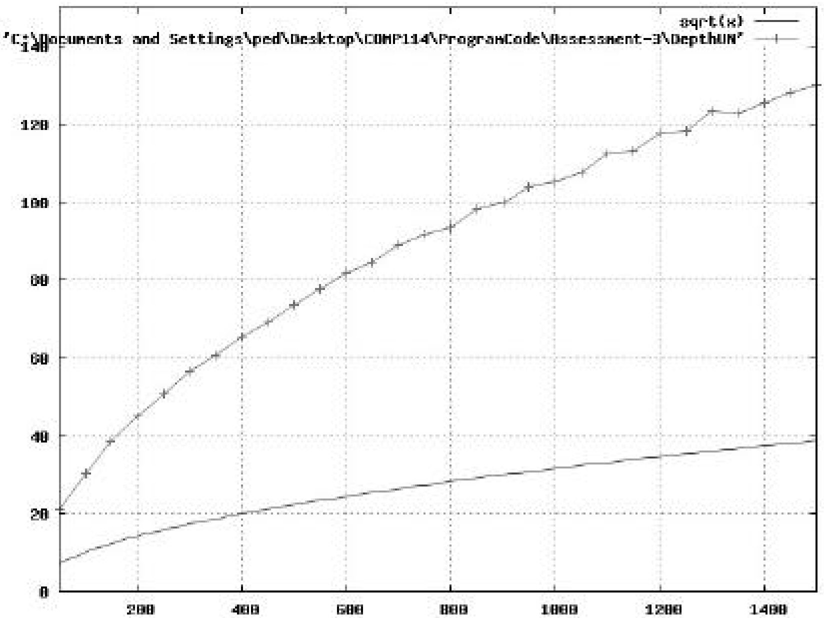
- Suppose we carry out the same process for the table of data produced by looking at the average depth of trees generated using the method UniformTree. That is,

1. Plot the curve formed by number of leaf nodes (x-axis) vs. average depth (y-axis).
2. Try to find a suitable function whose "shape" is "similar to" the resulting plot.

50	21.05	100	30.491
150	38.66	200	45.151
250	50.945	300	56.571
350	60.651	400	65.498
450	69.278	500	73.798
550	77.778	600	81.893
650	84.643	700	89.061
750	91.555	800	93.44
850	98.063	900	100.025
950	103.858	1000	105.438
1050	107.68	1100	112.416
1150	113.331	1200	117.651
1250	118.436	1300	123.25
1350	122.758	1400	125.508
1450	128.275	1500	130.074



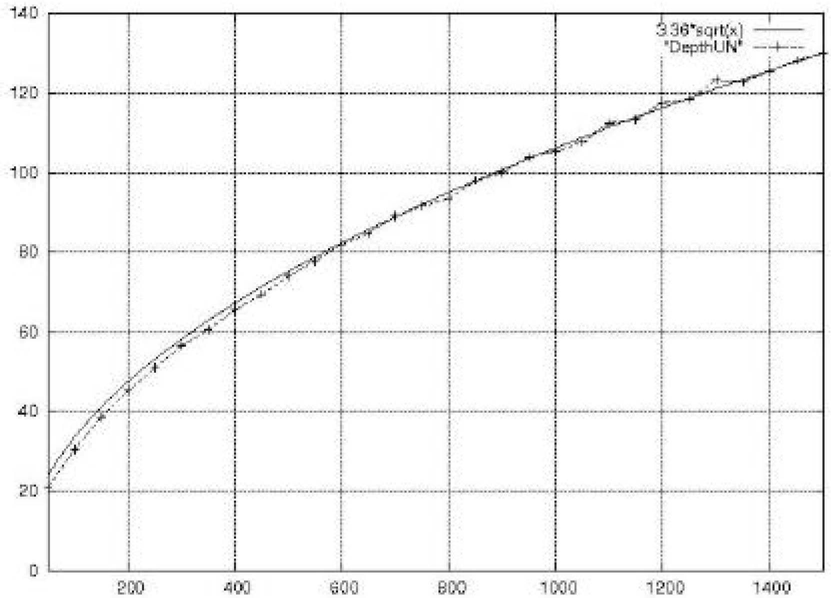
Possible comparison curve: $n^{0.5}$



15

Table produced by Depth/$n^{0.5}$

50	2.976919549	100	3.0491
150	3.156575782	200	3.192657828
250	3.222044708	300	3.266128208
350	3.241932317	400	3.2749
450	3.265796239	500	3.300346892
550	3.316465063	600	3.343267725
650	3.319971605	700	3.366189393
750	3.343115917	800	3.303602882
850	3.363533151	900	3.334166667
950	3.369598471	1000	3.334242319
1050	3.323076945	1100	3.389469931
1150	3.341948867	1200	3.396291826
1250	3.349875949	1300	3.418339959
1350	3.341049576	1400	3.354342395
1450	3.368666535	1500	3.358496238



Summary I

- In both cases the experiment results provide the bases for hypotheses about the behaviour of the random binary tree methods for arbitrarily many leaf nodes.
- For RootDown: $3.36\times\log(n)$
- For UniformTree: $3.36 \times n^{1/2}$.
- The value "3.36" is coincidental (a different figure would occur using base 2 logarithms).

Summary II – General method

- Given a table of result pairs <n, val(n)>, for a range of (evenly spaced) choices of n between LOW and HIGH (e.g. 50 and 1500 in steps of 50).
- Plot the curve of n (x-axis) against val(n) (y-axis).
- Choose a function – guess(x) – whose shape looks “close” to the curve resulting.
- Test how accurate this guess is by testing if val(n)/guess(n) “*appears to converge*” to some *constant value*.

Producing plotted curves

- A number of tools are available for displaying data in the form of plots.
- The program gnuplot is one such system. Although this uses a command line interface, recent releases offer window-based menus.
- Some basic gnuplot actions are described next.

Using gnuplot – some basic commands I

- set style data *keyword*

  where *keyword* is (normally) linespoints

  Data (from a file) is plotted with a straight line joining successive points.
- set output "*file-name*"

  All graphical output is placed in the file file-name. The file-name must be in " "
- set terminal *postscript*

  produce postscript output for viewing.

Using gnuplot – some basic commands II

- set xrange [low:high]

  set the x-axis to start at low and end at high. Similarly set yrange for y-axis.
- set grid

  Marks plotting area with a grid.
- plot *function(x)*

  produce the curve corresponding to function(x) on the output window/file.
- plot "*file-name*"

  plot the data file (formatted as a list of <x,y> pairs) specified by file-name.
- replot *function/file-name*

Example -

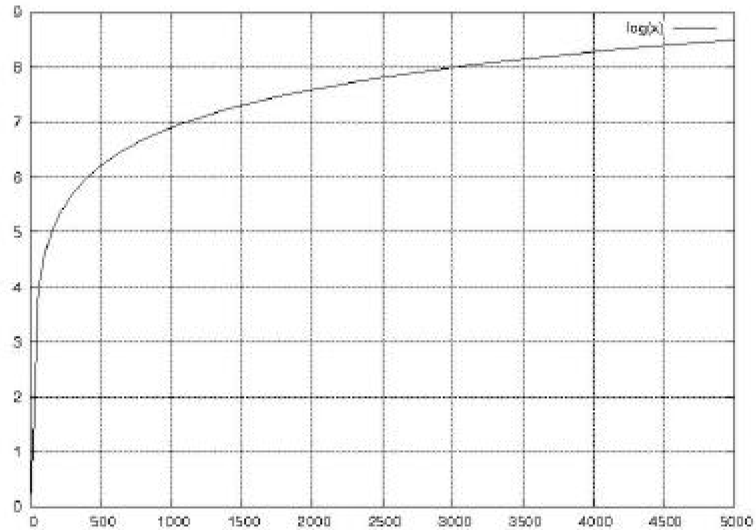
- The comparison between $3.36 \times n^{0.5}$ and the results of the UniformTree depth experiment was produced by

```
set style data linespoints
set xrange [50:1500]
set yrange [0:140]
set grid
plot 3.36*sqrt(x)
set output “compare.ps”
set terminal postscript
replot “DepthUN”
exit
```


Possible problems –

- In order to be effective some idea of which curves it will be reasonable to fit is needed.
- In the two examples we used the similarities between the “*data-curve*” and log(n) curve (for RootDown) and $n^{0.5}$ (for UniformTree).
- It helps to be familiar with some of the more commonly occurring cases that can appear in practical Computing applications.

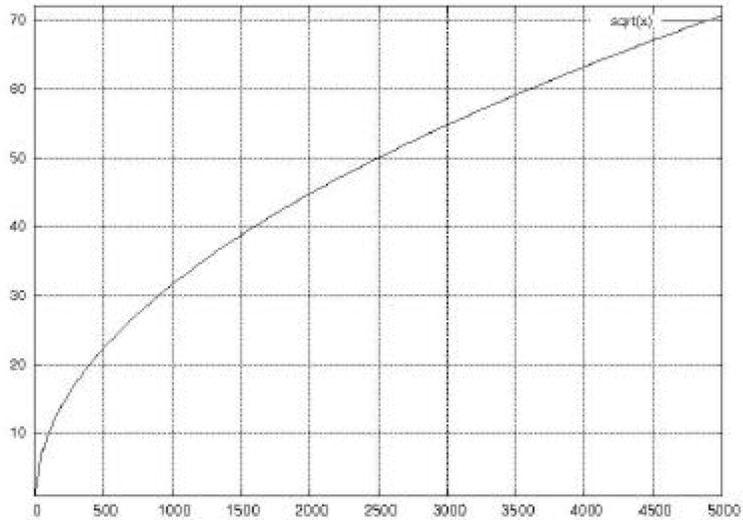
f(x) = log(x)



Features of f(x)=log(x) curve

- Although log(x) *exceeds* any given value once x is *large* enough, the *rate* at which it increases appears to *slow down*: after a “rapidly growing” start the curve itself becomes “*flatter*”.

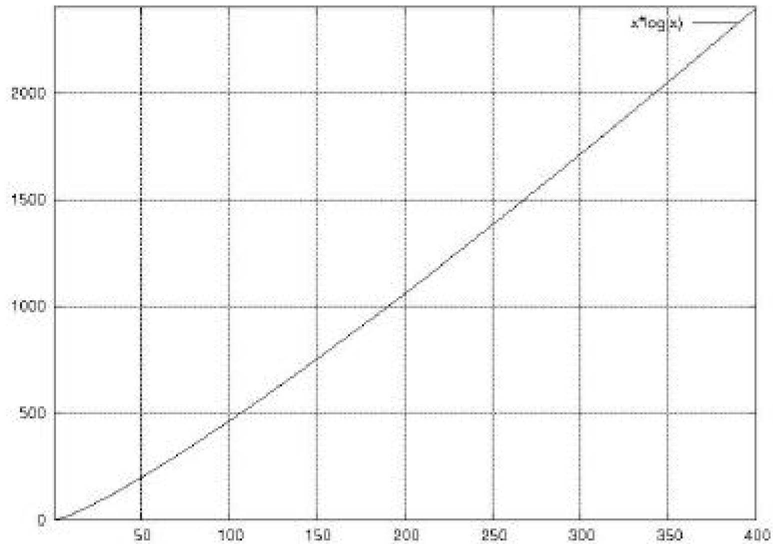
f(x) = sqrt(x)



Features of f(x)=sqrt(x) curve

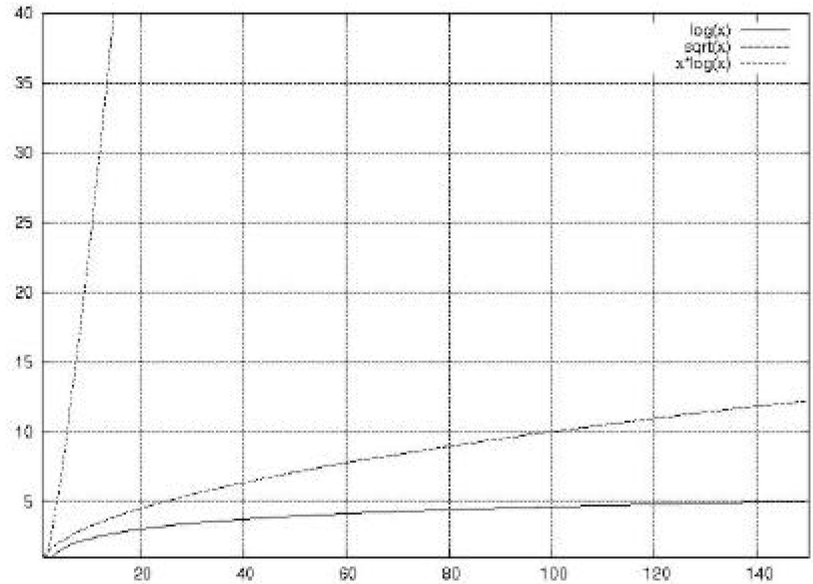
- The rate of growth is *significantly faster* than that of log(x).
- This curve "*appears*" to flatten out, however, the *distance* between log(x) and sqrt(x) will eventually *exceed* any fixed value, i.e. sqrt(x)-log(x) will be greater than any given number once x is large enough.

f(x) = x × log(x)

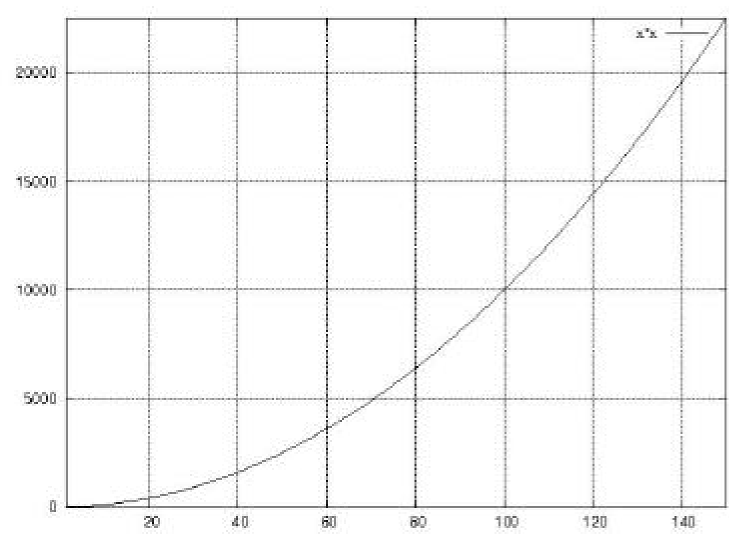


Features of f(x)=x × log(x) curve

- Rate of increase is much more rapid than either of the two preceding cases.
- The occurrence of x × log(x) in experiments as a "*rate of growth*" behaviour is quite *frequent*.
- Some examples ought to be found within the experimental investigation forming the final Assessment task.
- The next slide shows the three cases plotted on the same axes.

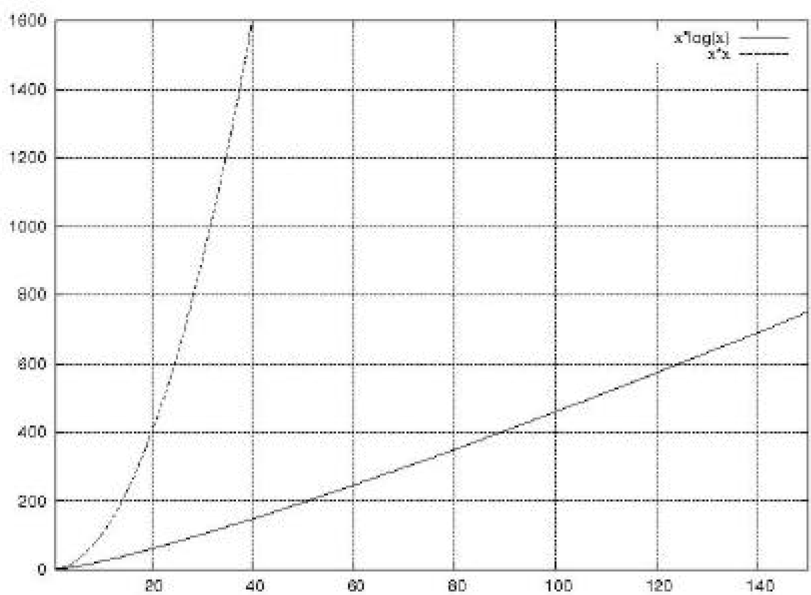

log(x)
sqrt(x)
x*log(x)
40
35
30
25
20
15
10
5
20
40
60
80
100
120
140

f(x) = x × x


x*x
20000
15000
10000
5000
0
20
40
60
80
100
120
140

Features of f(x)= x × x curve

- The curves produced by x, x×x, x×x×x … increase in value at progressively greater rates – x×x×x has a "*steeper*" curve than x×x which itself has a steeper curve than x.
- In "practical" contexts, "*useful*" program solutions will tend to show performance measuring x^k for a "*low*" value of k.



Summary

- The preceding examples give some indication of how *presentation* of experimental results may assist in arguing for an hypothesis.
- In both of the binary tree cases, the hypothesis "*the average depth of an n-leaf tree generated by* RootDown (UniformTree) *is* $c\times\log(n)$ ($d\times n^{0.5}$)" *can* be validated analytically (i.e. without recourse to experiments).
- In the UniformTree case, however, such analysis requires extremely sophisticated and advanced techniques.

Reporting and Presenting Experiments

- The form in which results are presented is one consideration in reporting an experiment's outcome.
- In total the structure of such reports ought to include all of the aspects that we now review.

Components in Experiment Report I

A. Introduction

a) *What were the aims of the experiment*?, e.g. "*to examine expected properties of randomly generated binary trees*"

b) *Why were experimental approaches used*?, e.g. "*some properties give insight into how effective different binary tree methods are*", "*certain properties are not easy to study by analytic techniques*", etc.

Components in Experiment Report II

B. Methodology

a) *How* were random data *obtained*?, e.g. *sampling* of user population, *program generated* (as in Binary Tree studies).

b) *Assumptions* and *expected behaviours* of random *data sets* used, e.g. that the sample population is *normally distributed*, that a random *bit/number generator* was "*suitable*", *biases* induced *in random objects* constructed, e.g. number of links in a random network.

Components in Experiment Report II

B. Methodology (continued)

c) Summary of *properties* of the data being considered, e.g. depth;

d) Description of the *statistical quantities computed*, e.g. *average* value of sample, *largest/smallest* value seen, standard deviation of sample.

e) What *range* of data sizes were used?

f) *How many trials* for each data size?

g) Possible *justifications* for choice of (e) and (f).

Components in Experiment Report III

C. Presentation of Experiment Results

Form of presentation: *Tables*; *Data plots* of individual outcomes and/or combined plots, e.g. as in showing UniformTree and RootDown results on same plot.

Components in Experiment Report IV

D. Analysis/Evaluation of Experiment Results

a) Do results *support* or *motivate new hypotheses*?, e.g. “the expected depth of random n-leaf trees approaches $c \times \log(n)$”

b) Arguments *justifying* such hypotheses, e.g. plots “*matching*” f(n) against data results; selected study of *predicted* behaviour with *larger* sizes.

Components in Experiment Report V

E. Conclusions

a) *Summary* of experimental *findings*, e.g. relationship between experimental *aims* and the *outcome*; extent to which findings are *inconclusive*; hypotheses motivated by experimental findings (if any).

b) Critical evaluation of *weak points* in experimental basis and solutions.