In several Computer Science labs, we ask you to analytically predict the running time of a procedure, and to experimentally confirm your prediction. You should make the analytical prediction by following the logic of the program, and *not* by taking timing data. You should be able to explain the reasoning behind your analysis clearly in writing.

For the experimental confirmation, you can tabulate or graph experimental results. The following table suggests good choices for what to tabulate or how to use graph paper to make a convincing case:

$T(n)$	Graphically	Using a table
$an + b$	Plot on regular graph paper, draw a straight line	Tabulate n, $T(n)$, and $T(n) - T(n - c) = ac$
ba^n	Plot using semi-log to get a line. Slope depends on a.	Tabulate $T(n)/T(n - 1) = a$
bn^a	Plot on log-log paper to get a line. Slope is equal to a.	Tabulate $T(2n)/T(n) = 2^a$
$a \log_b n$	Plot on semi-log with axes swapped to get a line.	Tabulate $T(2n) - T(n) = \log_b 2$

- To convince someone data fits a line, always draw a straight line with a ruler. Don't just connect the points.
- Also, be sure that you choose your axes so the slope of your line isn't too steep or too shallow.
- If the running times of two algorithms are being compared, be sure to include the data from both on the same table or graph so the reader can easily compare.

  This means that if an exponential time algorithm is being compared with polynomial time one, you may need to use two pieces of graph paper, and include both sets of data on at least one of the two papers.
- Don't forget to clearly label your axes, label the table columns, give your units. Be sure it's clear where the data came from (program A or program B.)