

NAME

gvgen – generate graphs

SYNOPSIS

gvgen [**-dv?**] [**-in**] [**-cn**] [**-C_{x,y}**] [**-g/f_{x,y}**] [**-G/f_{x,y}**] [**-hn**] [**-kn**] [**-b_{x,y}**] [**-B_{x,y}**] [**-mn**] [**-M_{x,y}**] [**-pn**] [**-r_{x,y}**] [**-R_x**] [**-sn**] [**-Sn**] [**-S_{n,d}**] [**-tn**] [**-td,n**] [**-T_{x,y}**] [**-T_{x,y,u,v}**] [**-useed**] [**-wn**] [**-nprefix**] [**-Nname**] [**-ooutfile**]

DESCRIPTION

gvgen generates a variety of simple, regularly-structured abstract graphs.

OPTIONS

The following options are supported:

- c n** Generate a cycle with n vertices and edges.
- C _{x,y}** Generate an x by y cylinder. This will have $x*y$ vertices and $2*x*y - y$ edges.
- g /f_{x,y}** Generate an x by y grid. If **f** is given, the grid is folded, with an edge attaching each pair of opposing corner vertices. This will have $x*y$ vertices and $2*x*y - y - x$ edges if unfolded and $2*x*y - y - x + 2$ edges if folded.
- G /f_{x,y}** Generate an x by y partial grid. If **f** is given, the grid is folded, with an edge attaching each pair of opposing corner vertices. This will have $x*y$ vertices.
- h n** Generate a hypercube of degree n . This will have 2^n vertices and $n*2^{(n-1)}$ edges.
- k n** Generate a complete graph on n vertices with $n*(n-1)/2$ edges.
- b _{x,y}** Generate a complete x by y bipartite graph. This will have $x+y$ vertices and $x*y$ edges.
- B _{x,y}** Generate an x by y ball, i.e., an x by y cylinder with two "cap" nodes closing the ends. This will have $x*y + 2$ vertices and $2*x*y + y$ edges.
- m n** Generate a triangular mesh with n vertices on a side. This will have $(n+1)*n/2$ vertices and $3*(n-1)*n/2$ edges.
- M _{x,y}** Generate an x by y Moebius strip. This will have $x*y$ vertices and $2*x*y - y$ edges.
- p n** Generate a path on n vertices. This will have $n-1$ edges.
- r _{x,y}** Generate a random graph. The number of vertices will be the largest value of the form 2^{n-1} less than or equal to x . Larger values of y increase the density of the graph.
- R x** Generate a random rooted tree on x vertices.
- s n** Generate a star on n vertices. This will have $n-1$ edges.
- S n** Generate a Sierpinski graph of order n . This will have $3*(3^{(n-1)} + 1)/2$ vertices and 3^n edges.
- S _{n,d}** Generate a d -dimensional Sierpinski graph of order n . At present, d must be 2 or 3. For d equal to 3, there will be $4*(4^{(n-1)} + 1)/2$ vertices and $6 * 4^{(n-1)}$ edges.
- t n** Generate a binary tree of height n . This will have 2^{n-1} vertices and $2^n - 2$ edges.
- t _{h,n}** Generate a n -ary tree of height h .
- T _{x,y}** Generate an x by y torus. This will have $x*y$ vertices and $2*x*y$ edges. If u and v are given, they specify twists of that amount in the horizontal and vertical directions, respectively.
- T _{x,y,u,v}** Generate an x by y torus. This will have $x*y$ vertices and $2*x*y$ edges. If u and v are given, they specify twists of that amount in the horizontal and vertical directions, respectively.
- u seed** Specify the initial state for the random number generator. If this option is not given, the current time is used to derive a seed.

- w *n*** Generate a path on *n* vertices. This will have *n-1* edges.
- i *n*** Generate *n* graphs of the requested type. At present, only available if the **-R** flag is used.
- n *prefix***
Normally, integers are used as node names. If *prefix* is specified, this will be prepended to the integer to create the name.
- N *name***
Use *name* as the name of the graph. By default, the graph is anonymous.
- o *outfile***
If specified, the generated graph is written into the file *outfile*. Otherwise, the graph is written to standard out.
- d** Make the generated graph directed.
- v** Verbose output.
- ?** Print usage information.

EXIT STATUS

gvgen exits with 0 on successful completion, and exits with 1 if given an ill-formed or incorrect flag, or if the specified output file could not be opened.

AUTHOR

Emden R. Gansner <erg@research.att.com>

SEE ALSO

gc(1), acyclic(1), gvpr(1), gvcolor(1), ccomps(1), sccmap(1), tred(1), libgraph(3)