

Graph Pebbling Interactive Exploration

By Ben Sauerwine

Download and run the pebbling workshop program from

<http://www.snakebyte.biz/Pebbling/pebbles.exe> - It is a standalone program under Windows 2000 and XP.

Unfortunately, the only way to be certain of a general nondirected graph's pebbling number seems to be to try every possibility. Consider the possibility of a probabilistic algorithm that would try to determine the pebbling number by trying just a few graphs.

If I were to create a path graph then randomly place $p-1$ pebbles on it (where p is the actual pebbling number), would you expect the arrangement to be a valid pebbling? How many invalid arrangements would there be?

What about a tree or cycle graph?

Based on the answers to the above questions, it should be clear that graph pebbling is not particularly vulnerable to a probabilistic algorithm. However, observe this:

Create a six-node graph with the connections however you like them. Find its pebbling number using the "brutal" feature of the workshop, then find how many pebbles you need to randomly place so that half of the random arrangements are valid. This algorithm seems more plausible if the proportion holds in general, or is a function of an easy-to-calculate measure of connectivity. Use the "Is this a valid pebbling" feature of the workshop. If you need a source of random numbers, use the `randInt(min, max)` function of your TI calculator.

Recall the data collected on minimum and maximum pebbling number versus the number of connections in the graph. If you need to see them again, check <http://www.snakebyte.biz/Pebbling/GraphPebbling.ppt>

What general shape or property did the minimum pebbling number graphs have?

What general shape or property did the maximum pebbling number graphs have?

Note that to get the graph pebbling number, one takes the maximum of the pebbling numbers with respect to each node. An algorithm could run dramatically faster, then, if it was possible to determine which node has the highest pebbling number easily.

Create several graphs with five or six nodes. Use the Brutal algorithm to determine the pebbling number with respect to each node. To reveal what each node's respective pebbling number is, click on it. Based on your layouts and the results, can you suggest an algorithm or heuristic to determine which node is the "worst-case" pebbling node?