Better than random: a hands-on introduction to quasirandomness

Jim Propp
U. Mass. Lowell
`propp@jamespropp.oh-ar-gee`

(based on articles in progress by Ander Holroyd, Lionel Levine, and Jim Propp)

All these slides are on the web at `jamespropp.org/PiMuEpsilon.pdf` so there's no need to take notes on anything you see here (only on the things that I say that you don't see!).

Puzzle #1: Each number between **1** *and* **5** *is equipped with a light, which can be green or red. A bug is dropped on* **3** *and obeys the following rules at all times: if it sees a green light, it turns the light red and moves one step to the right; if it sees a red light, it turns the light green and moves one step to the left.*

Show that the bug must eventually leave the system (either by exiting **1** *heading to the left, or by exiting* **5** *heading to the right), and give a simple rule for predicting which of the two outcomes will happen.*

Could the bug stay trapped in **1**, **2**, **3**, **4**, **5** forever, never escaping to "**0**" or "**6**"?

If so, there must be some site that the bug visits infinitely often (**3**, say).

But then it must visit **4** infinitely often (since half of the time when it leaves **3** it goes to **4**).

But then it must visit site **5** infinitely often (since half of the time when it leaves **4** it goes to **5**).

But on its first or second visit to site **5**, the bug must go off to the right!

Contradiction!

So the bug must eventually escape.

But which way will the bug escape?

The trick to figuring this out is to notice that the the quantity

(NUMBER OF GREEN LIGHTS)

plus

(POSITION OF THE BUG)

is *invariant*:

If a green light turns red (and the bug goes right), the number of green lights goes down by 1, but the position of the bug goes up by 1.

If a red light turns green (and the bug goes left), the number of green lights goes up by 1, but the position of the bug goes down by 1.

In particular, if the bug goes from **3** to **0** (that is, it leaves the system heading left), then number of green lights must increase by 3; this can't happen if the number of green lights was 3, 4, or 5 to begin with.

On the other hand, if the bug goes from **3** to **6** (that is, it leaves the system heading right), then number of green lights must decrease by 3; this can't happen if the number of green lights was 0, 1, or 2 to begin with.

So, the bug must exit to the right if the green lights outnumber the red lights, and to the left if the red lights outnumber the green lights.

Note that if you add a second bug to the system, it will exit the system on the opposite side. (If the first bug exited left, the second will exit right; if the first bug exited right, the second will exit left.)

If you add a third bug to the system, it will do the opposite of what the second bug did, that is, it will do the same as what the first bug did.

If you add lots of bugs to the system, one at a time, half of them will exit the system to the left and half will exit to the right (give or take half a bug).

Puzzle #2: What if the new bug is always added at **2** *instead of* **3***? How often (in the long run) will the bug exit to the left, and how often will it exit to the right?*

Count the green lights:

When the bug exits to the left, the number of green lights increases by $2-0=2$; when the bug exits to the right, the number of green lights decreases by $6-2=4$.

Old # of green lights:	0	1	2	3	4	5
New # of green lights:	2	3	4	5	0	1

So the number of green lights goes into a cycle, which is either $\ldots 0, 2, 4, 0, 2, 4, \ldots$ or $\ldots 1, 3, 5, 1, 3, 5, \ldots$

If you add lots of bugs to the system, one at a time, two-thirds of them will exit the system to the left and one-third will exit to the right (give or take a fraction of a bug).

Puzzle #3 (harder): Each positive integer on the number line is equipped with a green or red light. All lights are initially green. A bug is dropped on **1** *and obeys the following rules at all times: if it sees a green light, it turns the light red and moves one step to the right; if it sees a red light, it turns the light green and moves TWO steps to the left.*

Eventually the bug will fall off the line to the left, landing at either **-1** *or* **0**. *A second bug is then dropped on 1, then a third, and so on. Each successive bug that is added falls off the line to the left, landing at either* **-1** *or* **0**. *(This needs to be proved, but that's not the puzzle!)*

Show that the number of bugs that land at **-1***, divided by the number of bugs that land at* **0***, converges to* $\Phi = (1+\sqrt{5})/2 = 1.618...$*, the "golden ratio".*

Trick: Once again, you construct an invariant. This time you don't just count the lights; you treat the pattern of the lights as the representation of some number in "base Fibonacci"!

For more details, see Michael Kleber's article "Goldbug Variations" in the Winter 2005 issue of *The Mathematical Intelligencer*:
`people.brandeis.edu/~kleber/Papers/rotor.pdf`

If you add lots of bugs to the system, one at a time, $1/\Phi$ **of them will exit the system at -1 and** $1/\Phi^2$ **of them will exit the system at 0 (give or take a couple of bugs).**

Q. Why are these puzzles interesting mathematically?

A. They illustrate the way in which quasirandom walk mimics properties of random walk while cutting down on noise.

The building-blocks for these gadgets are called *rotor-routers*, which *route* the bugs in a *rotating* fashion.

Machines built out of rotor-routers are *deterministic*: their behavior does not involve any element of chance.

That is: you can predict in advance what they will do.

Let's compare these non-random systems with similar systems that are random.

E.g., for Puzzle #1, let's have each bug just choose randomly at each time-step whether to go right or left.

This is called a *random walk* model; such models are everywhere in modern science, from physics to finance.

Fact #1: If a bug starting from **3** does random walk on $\{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}\}$, where its chance of jumping one step to the left and its chance of jumping one step to the right are both equal to 1/2, then the bug has a 100% chance of eventually escaping, and its chance of escaping to the left and its chance of escaping to the right are both 1/2.

Fact #2: If a bug starting from **2** does random walk on $\{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}\}$, where its chance of jumping one step to the left and its chance of jumping one step to the right are both equal to 1/2, then the bug has a 100% chance of eventually escaping, and its chance of escaping to the left is 2/3 and its chance of escaping to the right is 1/3.

Fact #3: If a bug starting from **1** does random walk on $\{\mathbf{1},\mathbf{2},\ldots\}$, where its chance of jumping two steps to the left and its chance of jumping one step to the right are both equal to 1/2, then the bug has a 100% chance of eventually landing on **-1** or **0**; the chance of landing on **-1** is $1/\Phi$ and the chance of landing on **0** is $1/\Phi^2$. (Note: $1/\Phi + 1/\Phi^2 = 1$.)

Note that for each of the three puzzles, the way the bugs segregate (1/2 left versus 1/2 right; 2/3 left versus 1/3 right; and $1/\Phi$ at **-1** versus $1/\Phi^2$ at **0**) under **quasirandom** walk numerically agree with the probabilities under **random** walk.

Q. In what way is quasirandomness better than randomness?

A. The Law of Large Numbers "kicks in sooner".

For Fact #1, if we put N bugs through the random system, the number of bugs that escape to the left will be about $N/2$, with a typical error of about $\sqrt{N}$. But if we put N bugs through the non-random system of Puzzle #1, the number of bugs that escape to the left will be very close to $N/2$, with an error no greater than $1/2$.

Ditto for Fact #2 and Puzzle #2 (replacing "no greater than 1/2" by "no greater than 2/3").

Ditto for Fact #3 and Puzzle #3 (replacing "no greater than 1/2" by "no greater than Φ").

If we try to estimate $1/\Phi$ numerically using Fact #3, we could run N bugs through the random system and compute the number of **successes** divided by the number of **trials** (where a trial is successful if the bug goes to **-1** and a failure otherwise). The number of successes will be roughly $N \times 1/\Phi \pm \sqrt{N}$, so when we divide by N we'll get an estimate of $1/\Phi$ that's typically off by $\pm\sqrt{N}/N$ or $\pm 1/\sqrt{N}$.

That is, our estimate of $1/\Phi$ will have a built-in statistical error ("noise") on the order of $1/\sqrt{N}$. For instance, if we do a million random trials, our estimate of $1/\Phi$ will be accurate to within 1 part in 10^3 (three significant figures).

On the other hand, running a million bugs through the quasirandom system would give us **six** decimal digits of accuracy: $N \times 1/\Phi \ \pm \Phi$ divided by N gives $1/\Phi \pm \Phi/N$.

In this case, quasirandomization reduces our error from $1/\sqrt{N}$ to C/N for some small constant C, where N is the number of trials (bugs).

When N is large, C/N is much less than $1/\sqrt{N}$.

To see how Puzzle #3 goes, use the applet

`jamespropp.org/rotor-router-1.0/`

with the Graph/Mode selector set to "1-D Walk".

Set the Graph/Mode selector to "2-D Walk" to see a quasirandom gadget for approximating $\pi/8$.

(In summer 2006 I succeeded in proving that the error for this approximation decreases like $(\ln N)/N$ or better, where N is the number of trials.)

Set the Graph/Mode selector to "1-D Aggregation" to see a quasirandom gadget for approximating $\sqrt{2}$.

(When Lionel Levine was an undergraduate in 2002, he proved for his senior project that the error for this approximation decreases like $1/N$.)

Finally, set the Graph/Mode selector to "2-D Aggregation" to see a quasirandom gadget for growing circular blobs. See

`jamespropp.org/million.gif`

Lionel Levine and Yuval Peres proved in 2005 that these blobs really do become true circles in the limit. But the internal structures are still completely mysterious.

(This version of the rotor-router applet was created by University of Wisconsin undergrads Hal Canary and Francis Wong.)

Random walk is an example of a "Markov chain". In course 92.584, we'll study the theory of Markov chains, including some of the theory behind quasirandomness. See

`www.cs.uml.edu/~jpropp/584.html`

In EQL (Explorations in Quasirandomness at Lowell), students will be paid to serve as research interns in state-of-the-art research on quasirandomness, from 2007 to 2009. See

`www.cs.uml.edu/~jpropp/eql.html`