How Can Cryptographers with Alzheimer Locate Low Probability Peaks?

Itai Dinur, Orr Dunkelman,
Nathan Keller, Adi Shamir

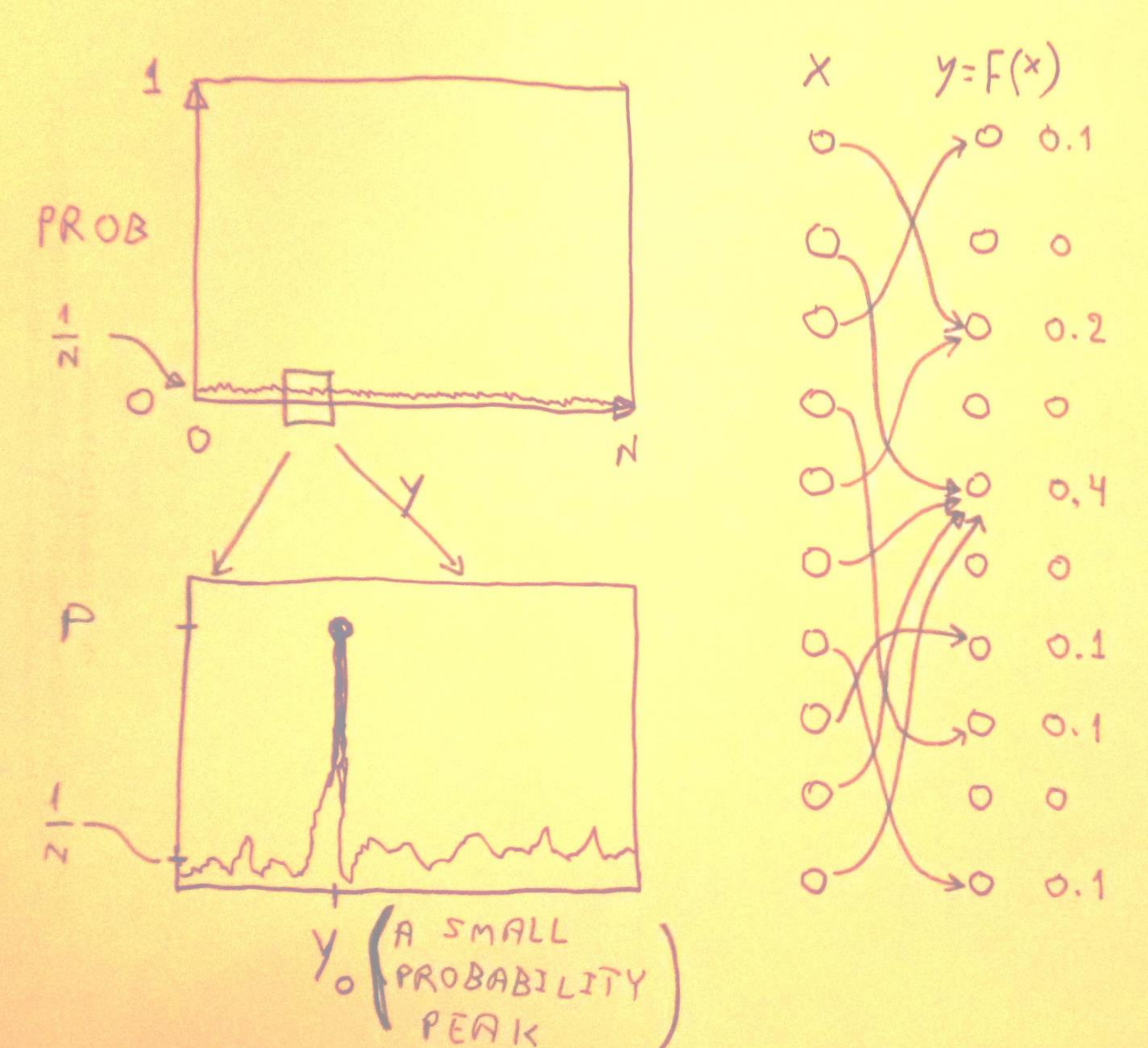
A Very Common Problem in Cryptography:

- You are given a random looking mapping F from n bit inputs x to n bit outputs y, in which most outputs occur with the expected probability of $p=2^{-n}$

- There exists some y_0 which occurs with higher probability $2^{-n} \ll p \ll 1$

- You want to locate this probability peak

Two Graphic Views of the Problem:



The Simplest Algorithm:

- Use a large array with $N=2^n$ counters:

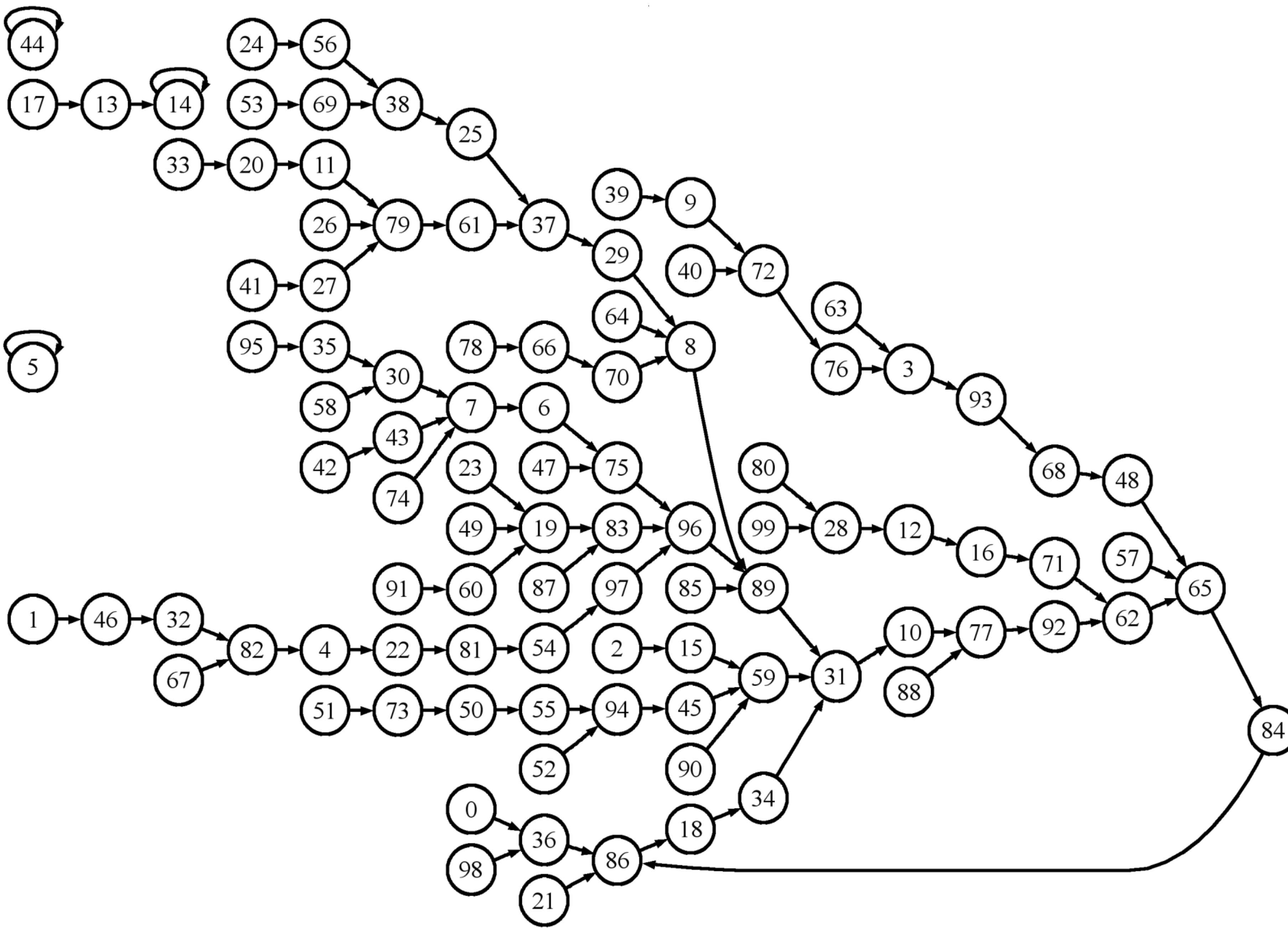
- Generate about $c*p^{-1}$ random outputs
- Count how many times each y was generated
- Most counters will contain either 0 or 1 occurrences
- Some counters will contain 2 due to birthday paradoxes
- The counter corresponding to y_0 will contain about c

0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	2	0	9	0	1

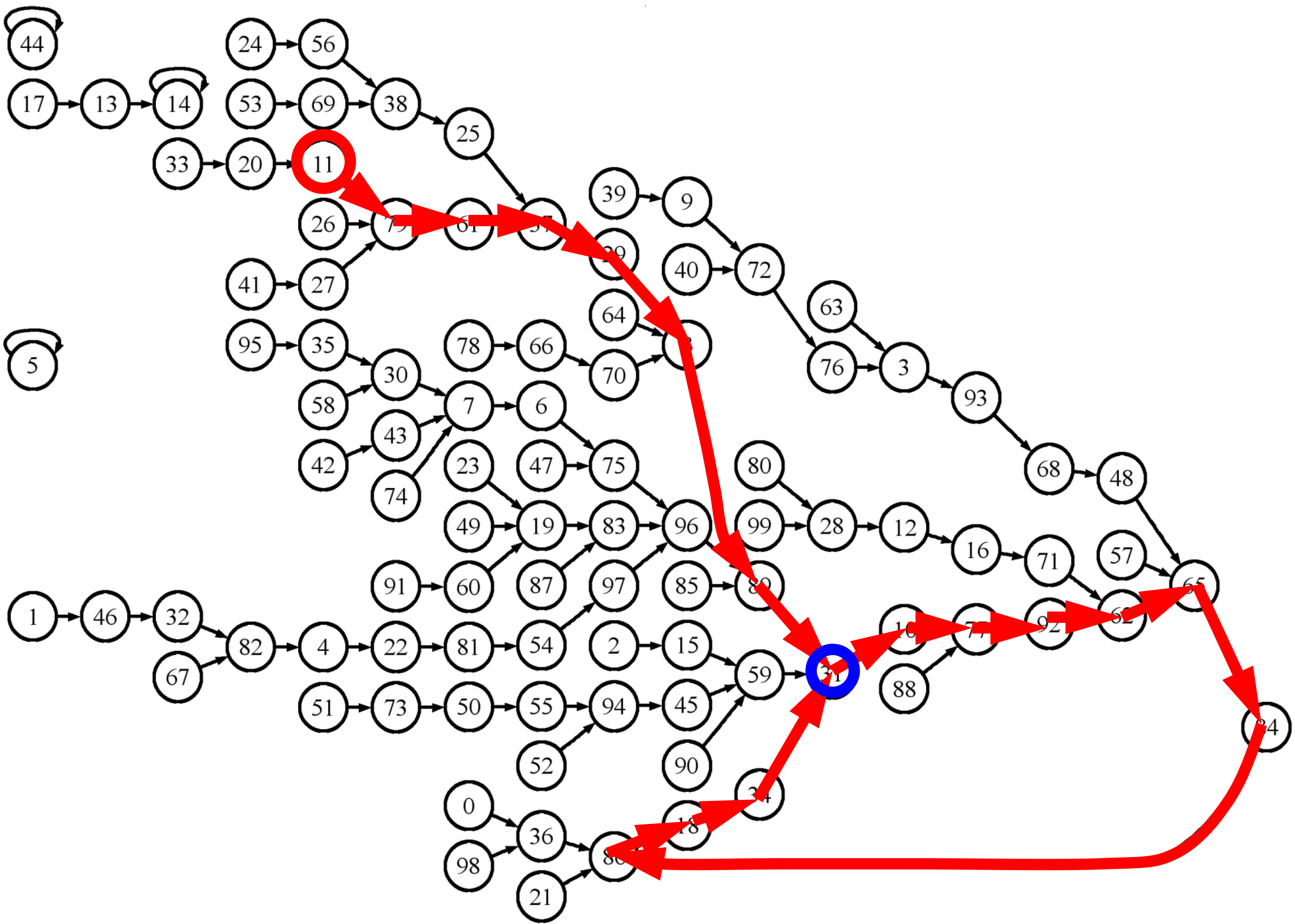
What Can We Do If We Have Alzheimer and Cannot Memorize Anything?

- Consider as a running example functions F whose inputs and outputs have n=80 bits
- Keeping N=2^{80} memory is too expensive and too slow
- Ideally, we would like to use a memoryless algorithm for this problem

Consider the Graph G Created by Iterating a Random F:



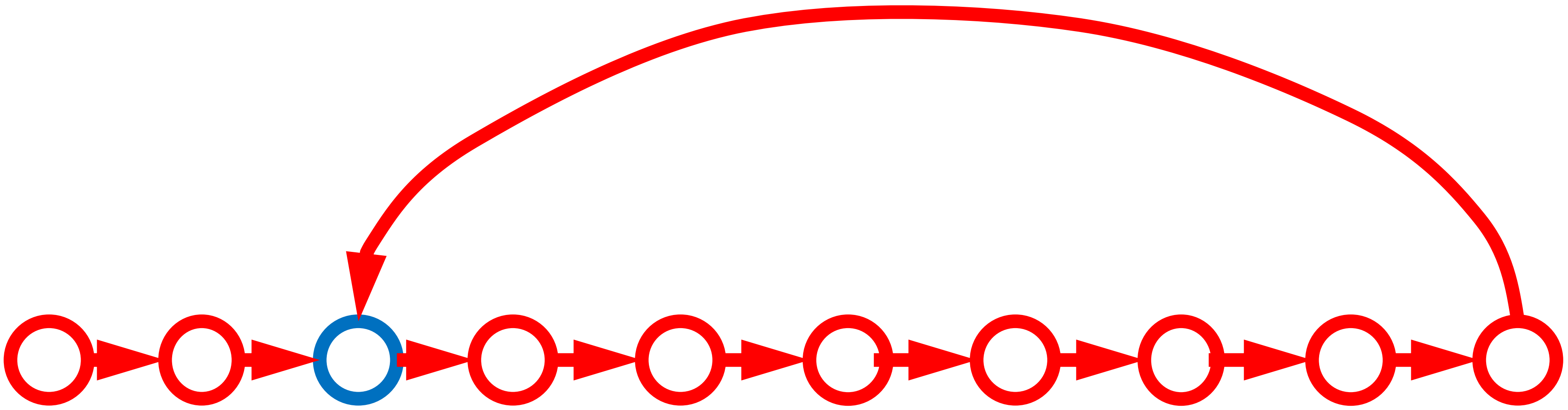
A Random Path in G and its Cycle Entry Point:



Pollard's Rho Method to Find the Cycle Entry Point:

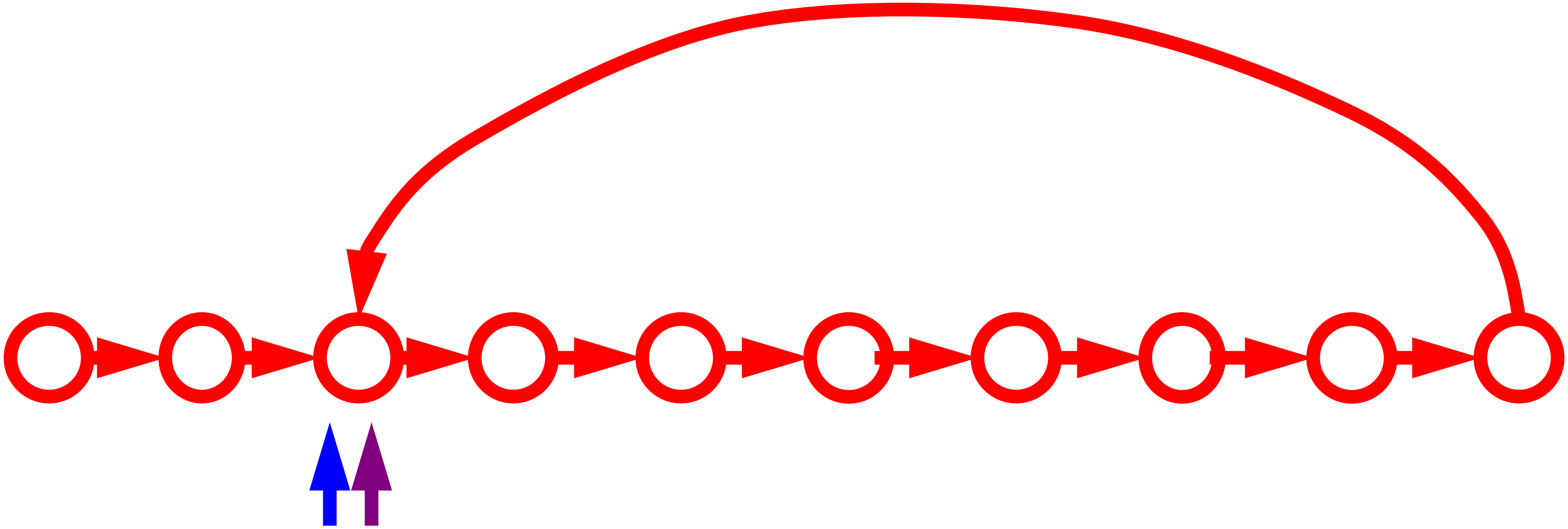
Uses constant memory (just two pointers)

Has a practical expected time of 2^{40}



What Happens When One Value y_0 in the Graph Has a Large Probability p>2^{-40}?

- This y_0 is likely to occur twice before any other point occurs twice by chance, so it is likely to be chosen as the cycle entry point



This Completely Solves the Problem of Finding Large Probability Peaks:

- When the probability of the peak y_0 is larger than 2^{-40}, we can find it in optimal time and no memory

- What can we do when $2^{-80}<p<2^{-40}$?

We Now Describe a New Memoryless Algorithm for Finding Low Probability Peaks between 2^{-40} and 2^{-60}:

- Closer inspection of Pollard's algorithm shows that the probability that y_0 will be the cycle entry point is higher than 2^{-40}, while the probability of a random y' to be the cycle entry point is about 2^{-80}
- **We can thus use a second Pollard process to identify this new high probability peak!**

Using Multiple Flavors of F:

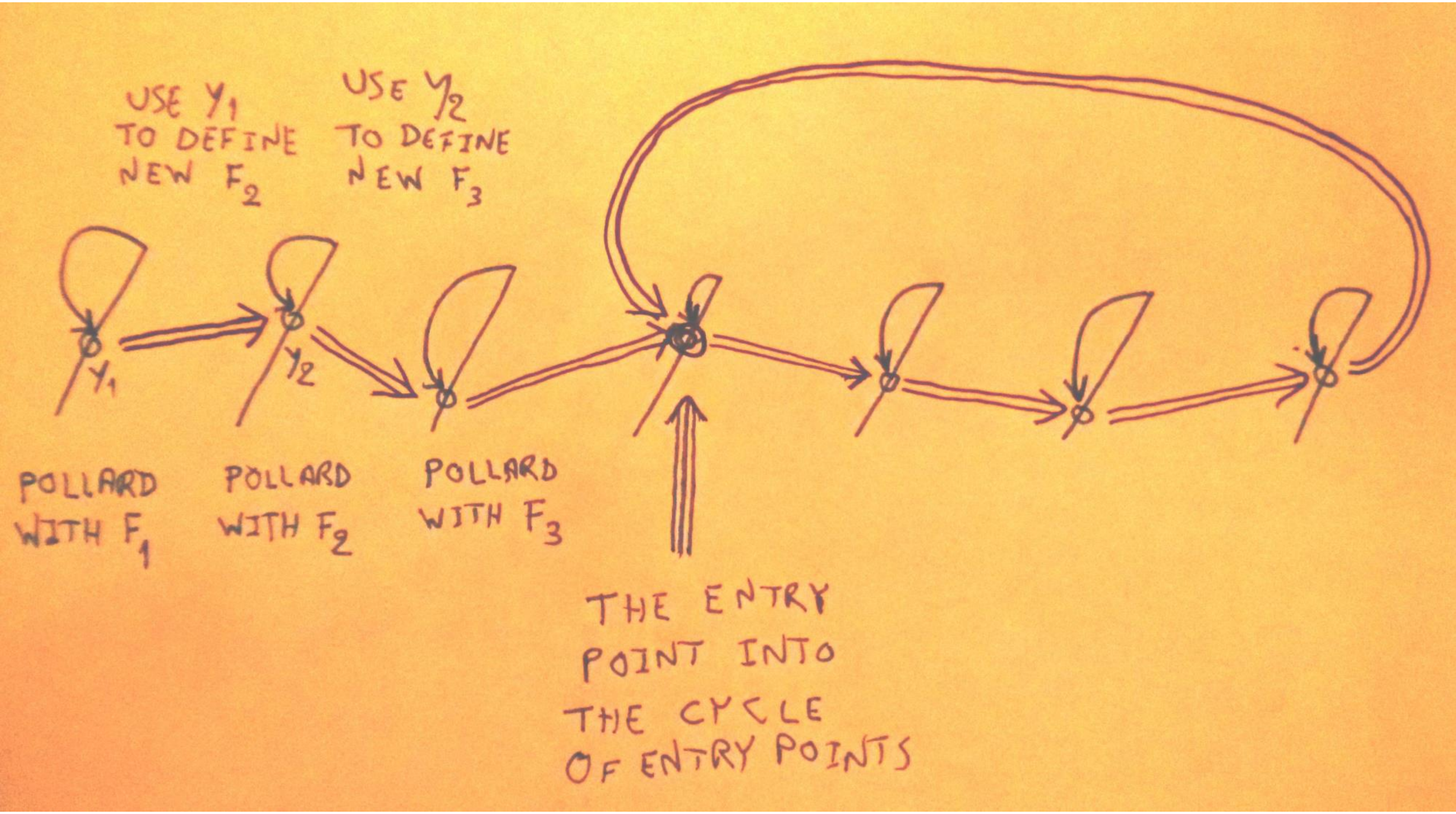
(The main idea in Hellman's time/memory tradeoff)

- Define the i-th flavor of $F(x)$ as the new function

  $F\_i(x)=F(x+i)$

- It completely changes the global structure of G
- It keeps many of the local properties of G
- **In particular, a popular value remains popular!**

How the New Pollard^2 Finds Low Probability Peaks:



Pollard^2 Experimental Data:

N = 2^28					
Prob. of the high value relatively to N	Prob. of the high value (in log_2)		Total trials	High value found	Percent
N^-0.5	-14	14	100	100	100.00%
N^-0.54	-15	15	100	100	100.00%
N^-0.57	-16	16	100	100	100.00%
N^-0.61	-17	17	100	97	97.00%
N^-0.64	-18	18	100	91	91.00%
N^-0.68	-19	19	100	71	71.00%
N^-0.71	-20	20	100	32	32.00%
N^-0.75	-21	21	100	8	8.00%
N^-0.79	-22	22	100	0	0.00%

In the Full Paper
(which will soon appear on ePrint):

- We describe many extensions and optimizations of the new peak finding algorithm:
  - How to find multiple probability peaks of various heights
  - How to find even lower peaks with probability p below 2^{-3n/4} in optimal time using a small amount of memory
  - We describe various time/memory tradeoffs for this important problem

THANK YOU!!!