Collision algorithms.

Our next goal is to study a generalization of Shanks' algorithm. We will need to discuss a few problems in probability theory, which are of some interest in their own right.

The birthday paradox.

(1) What is the probability that out of a random group of n people someone has birthday on a given day?

(2) What is the probability that at least two people share the same birthday?

(assume the year has 365 days)

Answers:

(1) It is easier to compute the probability that noone has his/her birthday on the given day (your birthday) and use that the sum of probabilities of complimentary events is 1:

$$P(A) + P(A^c) = 1.$$

$P\left(\begin{array}{c}\text{noone has birthday}\\ \text{on the given day}\end{array}\right) = \left(\frac{364}{365}\right)^n$ (the birthdays are independent and any day, except the given one, works).

Hence, we get $1 - \left(\frac{364}{365}\right)^n$.

(2) Again, using similar logic,

$$P\left(\begin{array}{c}\text{two people have}\\ \text{the same birthday}\end{array}\right) = 1 - P\left(\begin{array}{c}\text{all } n \text{ people have}\\ \text{different birthdays}\end{array}\right) =$$

$$= 1 - \underbrace{\frac{365}{365}}_{\substack{\text{no restr-ns}\\ \text{on the birthday}\\ \text{of the } 1^{st} \text{ person}}} \cdot \underbrace{\frac{364}{365}}_{\substack{\text{the second person}\\ \text{can have a birthday}\\ \text{on any day except the}\\ \text{first person's birthday...}}} \cdot \frac{363}{365} \cdot \ldots \cdot \frac{365-(n-1)}{365}$$

Example. My birthday is July 4^{th} (it actually is :)). We have 15 students in the class. What is the probability that someone has birthday on 07/04 as well?

Answer: $1-\left(\frac{364}{365}\right)^{15} \approx 0.04$ (4%).

The probability that two of you have birthdays on the same day is

$$1-\frac{365}{365}\cdot\frac{364}{365}\cdot\ldots\cdot\frac{351}{365} \approx 0.25 = 25\%$$

The second answer suggest that the probability of the corresponding event is a lot higher than that of the first one. As the events look similar at a first glance, but their probabilities are very different, such an instance is reffered to as paradox.

Thm. An urn contains N balls (red and blue), n are red and $N-n$ are blue. We randomly pick a ball from the urn, record its color and put it back. The procedure is repeated m times.

(a) The probability that at least one red ball will be picked is $P(n \geq 1) = 1-\left(1-\frac{n}{N}\right)^m$.

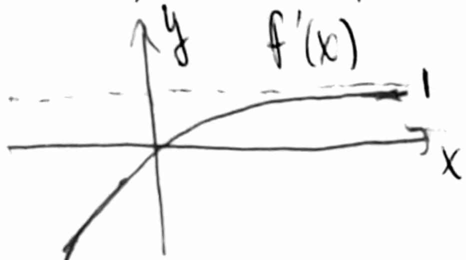
(b) The lower bound is

$$P(n \geq 1) \geq 1-e^{-\frac{mn}{N}}$$

Proof: (a) very similar to the 'birthday paradox' (think it through!).

(b) We first show that $e^{-x} \geq 1-x$ for $\forall x \in \mathbb{R}$.
Consider the function $f(x) := e^{-x} - 1 + x$. We need to show that $f(x) \geq 0$ for all $x \in \mathbb{R}$.

Notice that $f(0) = 1 - 1 + 0 = 0$. It is sufficient to check that $f(x)$ decreases for $x < 0$ and increases for $x \geq 0$. As $f'(x) = -e^{-x} + 1 \geq 0$ for $x \geq 0$ and $f'(x) \leq 0$ for $x \leq 0$, the assertion follows.



Using the inequality for $x = \frac{n}{N}$, we get

$$e^{-\frac{n}{N}} \geq 1 - \frac{n}{N} \Rightarrow e^{-\frac{nm}{N}} \geq \left(1 - \frac{n}{N}\right)^m \Leftrightarrow -e^{-\frac{nm}{N}} \leq -\left(1-\frac{n}{N}\right)^m$$

x^m is monotonically increasing

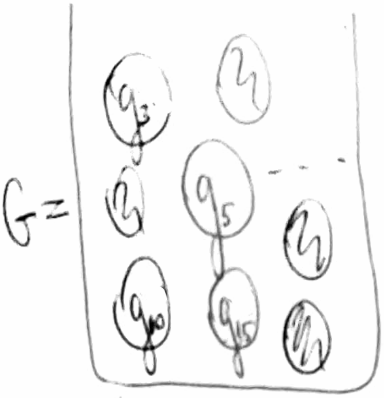
$$\Leftrightarrow 1 - \left(1 - \frac{n}{N}\right)^m \geq 1 - e^{-\frac{nm}{N}}.$$

$\boxed{}$

Application: let G be a finite group and $L_1, L_2 \subset G$ two subsets (not necessarily subgroups) of cardinality $n < N$. Then the probability that the intersection of L_1 and L_2 is nonempty satisfies the inequality

$$P(L_1 \cap L_2 \neq \emptyset) \geq 1 - e^{-\frac{n^2}{N}}$$

(we assume that the el-ts of L_1 and L_2 are chosen randomly).



The probability that the intersection of L_1 and L_2 is nonempty is the same as probability that one of elements in L_2 is in L_1 as well. The latter is equal to the probability of drawing at least one red ball (el-t of L_1) in n (cardinality of L_2) tries. Hence, it satisfies the inequality

$$P(L_1 \cap L_2 \neq \emptyset) \geq 1 - e^{-\frac{n^2}{N}}.$$

For instance, if $n = \lceil \sqrt{N} \rceil$, then $P(L_1 \cap L_2 \neq \emptyset) \geq 1 - e^{-1} \approx 63.2\%$ while if $n = \lceil 3\sqrt{N} \rceil$, then $P(L_1 \cap L_2 \neq \emptyset) \geq 1 - e^{-9} \approx 99.9877\%$.

<u>Remark</u>. Unlike the setup of Shanks' algorithm, we are not guaranted that the lists

$$L_1 = \{g_1, g_2, \dots, g_n\} \text{ and } L_2 = \{S_1, S_2, \dots, S_n\}$$

(for an arbitrary group G)
have a nonempty intersection.

<u>Collision algorithm for DLP.</u>

<u>Proposition</u>. Let G be a group and g an element of order N. Assume we have a well-defined DLP, i.e. $g^s = h$ (with h being given). Then the solution (s) can be found in $O(\sqrt{N})$ steps (each step is exponentiation in G), using the collision algorithm.

Proof. We mimic the approach in Shank's algorithm. Write $s = k - m$ and look for a solution of DLP in the form

$$g^k = h \cdot g^m$$

We make a random choice of n numbers $\{a_1, \dots, a_n\}$ and set the first list to be $L_1 = \{g^{a_1}, g^{a_2}, \dots, g^{a_n}\}$. Next, randomly pick another n-tuple $\{b_1, b_2, \dots, b_n\}$ and set

$$L_2 = \{h \cdot g^{b_1}, h \cdot g^{b_2}, \dots, h \cdot g^{b_n}\}.$$

The 'urn' S will be the set $\{1, g, g^2, \dots, g^{N-1}\}$. Notice that $h = g^s$ implies $L_1, L_2 \subset S$ (L_1 and L_2 are subsets of S).

Once again, noone can guarantee $L_1 \cap L_2 \neq \emptyset$, but for n sufficiently large (say, $n \sim 2\lceil\sqrt{N}\rceil$ or $3\lceil\sqrt{N}\rceil$) the probability $P(L_1 \cap L_2 \neq \emptyset)$ is very close to 1.

Let $x \in L_1 \cap L_2$, then $x = g^k = h \cdot g^m$ (for some $k \in \{a_1, a_2, \dots, a_n\}$ and some $m \in \{b_1, \dots, b_n\}$).

Then $s \equiv k - m \pmod{N}$ is the solution to DLP.

Pollard's ρ method.

Goal: using the collision type algorithms requires storing a lot of numbers ($\sim 2\sqrt{N}$ as $|L_1| = |L_2| \approx \sqrt{N}$), which becomes an issue for large values of N. The algorithm invented by Pollard allows to bypass this issue.

Let S be a finite set and $f: S \to S$ a function. Suppose we choose a point $x = x_0 \in S$ and iteratively apply f to it, thus getting the sequence

$$x_0 = x$$

$$x_1 = f(x)$$

$$x_2 = f(x_1) = f \circ f(x)$$

$$\vdots$$

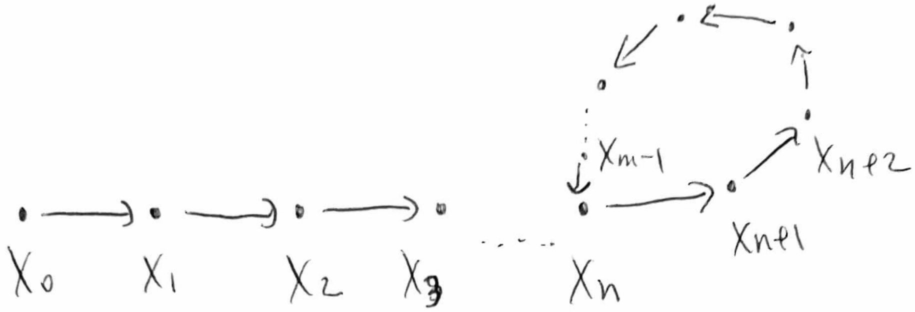
$$x_n = \underbrace{f \circ f \circ \ldots \circ f}_{n}(x) = f^{\circ n}(x)$$

$$\vdots$$

of points in S.

Rmk. The pair (S, f) is called a discrete dynamical system.

Rmk. As S is finite, there must exist a pair of indices (m, n), $m > n$, such that $x_m = x_m$.



<u>Notation:</u> $T := \min_n \{n \mid \exists m : x_m = x_n\}$ (tail)

$L :=$ length of the loop $= \min_n \{n \mid x_{T+n} = x_T\}$.

<u>Rmk.</u> The shape of the picture above resembles ρ, hence, the name of the method.

We introduce one more sequence:

$$y_0 = x_0$$
$$y_1 = f \circ f(x) = x_2$$
$$y_2 = f^{\circ 4}(x) = x_4$$
$$\vdots$$
$$y_n = x_{2n}$$
$$\vdots$$

<u>Obvious (but important) lemma.</u>

$$x_j = x_i \iff \begin{cases} i \geq T \\ j \equiv i \pmod{L} \end{cases}$$

<u>Corollary.</u> $y_i = x_i \iff \begin{cases} i \geq T \\ i \equiv 0 \pmod{L} \end{cases}$.

$(j = 2i - i = i)$.

Thm (Pollard's ρ Method). Let S be a finite set, $|S|=N$, $f: S \to S$ a map, $x \in S$ a chosen 'starting element'.

(a) If the sequence $\{x_0, x_1, x_2, \dots\}$ has a tail of length T and loop of length L, then

$$x_{2i} = x_i \quad \text{for some } 1 \le i \le T+L.$$

(b) If the map f is sufficiently random, then the mean value(*) of min. i, satisfying (a) is

$$\mathbb{E}(i) \approx 1.25\sqrt{N}.$$

(*): among different choices of the starting point.

Rmk. At each moment in time we only need to store x_i and $y_i = x_{2i}$.

Example: DLP for $\mathbb{F}_p^*$.

Let's see how to make Pollard's algorithm work in case $S = \mathbb{F}_p^\times$ and how it helps for solving (reducing the storage) the DLP: $g^s \equiv h \pmod p$

Step 1. We choose the function

$$f(x) := \begin{cases} g \cdot x, & 0 \le x < p/3 \\ x^2, & p/3 \le x < 2p/3 \\ hx, & 2p/3 \le x < p \end{cases}$$

Rmk. It is not known if $f(x)$ is sufficiently random! But experimentally ok.

We start with the element $x_0 = x = 1$, so
$$x_n = f^{\circ n}(1) = g^{\alpha_n} \cdot h^{\beta_n} \text{ with}$$

$$\alpha_{n+1} = \begin{cases} \alpha_n + 1, \\ 2\alpha_n \\ \alpha_n \end{cases} \quad \beta_{n+1} = \begin{cases} \beta_n & 0 \leq x_n < p/3 \\ 2\beta_n & p/3 \leq x_n < 2p/3 \\ \beta_n + 1 & 2p/3 \leq x_n \leq p \end{cases}$$

Similarly, $y_0 = x_0 = 1$, $y_n = g^{\gamma_i} \cdot h^{\delta_i}$ and the collision $y_n = x_{2n}$ implies $g^{\alpha_n} \cdot h^{\beta_n} = g^{\gamma_n} \cdot h^{\delta_n}$ or

$(*)$ $g^u \equiv h^v \pmod p$, where $u \equiv \alpha_n - \gamma_n$ and $v \equiv \delta_n - \beta_n \pmod{p-1}$.

The congruence $(*)$ implies $u \equiv v \log_g h \pmod{p-1}$. In case $\gcd(v, p-1) = 1$, we have that v is invertible modulo $p-1$ and $\log_g h \equiv u \cdot v^{-1} \pmod{p-1}$.

The case $\gcd(v, p-1) \geq 2$ requires a bit more work (see page 241 in the book). Also see page 242 for a concrete example (with actual numbers).