3.3 MILLER's Primality Test

How can we exploit the criterion for strong pseudoprimes to sufficiently many bases and construct a practically usable test? First we formulate the algorithm for one base a and assess its cost.

Since we anyway compute a^{n-1} by the binary power algorithm it makes sense to compute the complete sequence of powers beginning with a^r in a passing strike. Then the effort is about the same as for the "weak" pseudoprime test alone. Thus the test for strong pseudoprimes to the base a runs as follows:

Procedure sPPT(a)
[Strong pseudoprime test to base a]
Input parameters:
n = the integer to be tested (odd ≥ 3)
a = base (in the integer interval $[2 \ldots n-1]$)
Output parameters:
compo = a Boolean value with the meaning
TRUE: n is composite.
FALSE: The test has no definite result
[i. e. n is a strong pseudoprime to base a].
Instructions:
Compute s = 2-order of $n-1$.
Compute r = odd part of $n-1$.
Compute $b = a^r \bmod n$ (using the binary power algorithm).
Set $k = 0$.
[Loop: entry condition $b = a^{2^k r} \bmod n$
The Boolean variable 'done', initiated with FALSE, decides about repeating the loop.]
While not done:
If $b = 1$: set done = TRUE.
If $k = 0$: set compo = FALSE,
else: set compo = TRUE. [1 without preceding -1]
If $b = n-1$ and $k < s$:
set compo = FALSE, done = TRUE.
If $k = s$ and $b \neq 1$:
set compo = TRUE, done = TRUE.
In all other cases $[k < s, b \neq 1, b \neq n-1]$
replace b by $b^2 \bmod n$,
replace k by $k+1$.

To assess the cost we break the procedure down into single steps that each multiply two integers mod n. Computing $a^r \bmod n$ takes at most $2 \cdot \log_2(r)$ steps. In each of the up to s loops we compute a square. Since

$\log_2(n-1) = s + \log_2(r)$ we have to compute at most $2 \cdot \log_2(n)$ products $\bmod n$. Each of these squares needs at most N^2 "primitive" integer multiplications where N is the number of places of n in the used representation of the number system. Computing r takes s divisions by 2 that can be neglected. Hence a coarse estimate of the total cost yields $\mathrm{O}(\log(n)^3)$ for a single base.

MILLER's primality test is the sequence of strong peudoprime tests to the bases $2, 3, 4, 5, \ldots$. This doesn't look efficient: In the worst case we test a true prime, then we run through all bases $< n$. However as MILLER showed, significantly less bases suffice—*presupposed that the extended* RIEMANN *hypothesis is true.* In the next section we'll see some explanation but without complete proofs.