- Fermat's little theorem: If n is prime, then for every $1 \leq a < n$, $a^{n-1} \equiv 1 \mod n$. — (1)

  Contrapositive of Fermat's little theorem: For some a ($1 \leq a < n$), if $a^{n-1} \not\equiv 1 \mod n$ then n is not a prime. — (2)

- Algorithm: Pick a positive integer $a \in [1, n)$ at random; If $a^{n-1} \equiv 1 \mod n$ (i.e., passed the Fermat's test) then output 'n is prime' otherwise, output 'n is composite'. — (3)

  From (2), there is a possibility of an error only when algo outputs 'n is prime'; this error probability is upper bound underneath (after introducing a few definitions). And, the worst-case time complexity of the algorithm is upper bounded. Hence, it is a Monte Carlo algorithm with one-sided error.

- A composite integer n is called a *(Fermat) pseudoprime to base a* whenever $a^{n-1} \equiv 1 \mod n$ for any integer $a > 1$

  A composite integer n is called an *absolute psudoprime (a.k.a. a Carmichael number)* whenever $a^{n-1} \equiv 1 \mod n$ for every integer $a \in [1, n)$ with $gcd(a, n) = 1$.

  A side note of above definition: If a composite integer n is not an absolute pseudoprime, then there exists an integer $1 \leq b < n$ such that $gcd(b, n) = 1$ and $b^{n-1} \not\equiv 1 \mod n$.

- Assuming composite integer n is not an absolute pseudoprime (i.e., there exists an integer b such that $1 \leq b < n$, $gcd(b, n) = 1$, and $b^{n-1} \not\equiv 1 \mod n$), there are at least as many integers in $[1, n)$ that fail the Fermat's test as the number of integers in that range that pass the Fermat's test.

  * The proof of this theorem consists of two parts: (i) every $a < n$ that passes Fermat's test w.r.t. n has a twin $a.b \mod n$ that fails the test, (ii) in addition, every such $a \cdot b \mod n$, for fixed b but different choices of a, are distinct.

  Corollary: Assuming n is not an absolute pseudoprime, the probability n passes the Fermat's test (i.e., $a^{n-1} \equiv 1 \mod n$) for any randomly chosen $a \in [1, n)$ with $gcd(a, n) = 1$ is upper bounded by $\frac{1}{2}$. — (4)

- Under the assumption that the input n to algorithm (refer to (3)) is not an absolute pseudoprime, from (4), the probability of error when this algorithm outputs 'n is prime' is at most $\frac{1}{2}$.

- To reduce the error probability, apply the *abundance of witnesses* design paradigm: Do the Fermat's test k times, each time by choosing a from $[1, n)$ uniformly at random, output 'n is a prime' only if every a chosen passes the Fermat's test. This reduces the upper bound on the error probability to $\frac{1}{2^k}$, and the time complexity of this algorithm is weakly-polynomial.

[1] note by R. Inkulu, http://www.iitg.ac.in/rinkulu/