# PAL labs 10

23 / 11 / 2022

It is easy to generate random numbers from the $\{1, 2, 3, 4, 5, 6\}$ by throwing a dice. Suppose we have only one dice and we have to generate random integers in the interval $[0, 10]$. Describe the strategy of dice throwing which will generate each integer 0, 1,..., 10 with the same probability. (The dice is a classical 6-sided one).

There is an array of sorted integer values. Describe a strategy which will rearrange the values into a random order using a pseudorandom number generator. The method should work in a time proportional to the length of the array.
By rearranging into a random order we mean that all possible permutations of the values are equally likely.

Find out whether the length of the period of the given linear congruential generator is maximum possible.

- A) $x_{n+1} = (91x_n + 49) mod\ 600$
- B) $x_{n+1} = (8x_n + 80) mod\ 49$
- C) $x_{n+1} = (37x_n + 55) mod\ 144$
- D) $x_{n+1} = (99x_n + 81) mod\ 113$

10/5. Determine the period length in output of the Lehmer generator given by the relation $x_{n+1} = ((M-1) * x_n) \ mod \ M$, ($M$ is a prime).

10/6. Determine the upper and the lower bound of number of primes in the interval

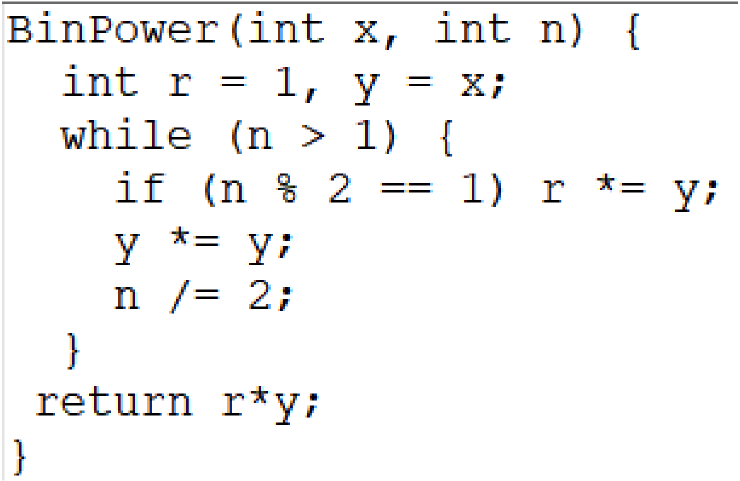
- A) $[0, 10^9]$
- B) $[10^9, 2 * 10^9]$
- C) $[2 * 10^9, 3 * 10^9]$

10/7. We say that an integer as a quasi-prime if it is an integer power of a prime. Write a pseudo-code of a modification of Eratosthenes' sieve which will generate exactly all quasi-primes.

A set $S = \{1000, 1001, ..., 999999\}$ was originally given. Then, all multiples of all primes less then 1000 $(2, 3, 5, \ldots, 991, 997)$ were excluded from $S$. Give an estimate of the cardinality of $S$ and of the number of primes in $S$.

Determine the maximum number of primes in any of the intervals $[30k, 30k + 29]$, $k = 1, 2, 3, 4, \ldots$ .

The given code calculates integer power $x^n$. Modify the code in such way that it will calculate $x^n \ mod \ m$, for positive integer $m$.



```
BinPower(int x, int n) {
  int r = 1, y = x;
  while (n > 1) {
    if (n % 2 == 1) r *= y;
    y *= y;
    n /= 2;
  }
 return r*y;
}
```