

```

binaryStringsNo00s = z1*. (z0 .z1+)*. (z0 + ε)
z1*. (z0 .z1+)*. (z0 + ε)

ogf[ε] := 1
ogf[a_Symbol] := z;
ogf[a_ + b_] := ogf[a] + ogf[b];
ogf[a_ .b_] := ogf[a] * ogf[b];
ogf[a_ ^k_] := ogf[a]^k;
ogf[a_*] := (1 - ogf[a])^-1;
ogf[a_^] := ogf[a] ogf[a*];
OGF[exp_] := FullSimplify[ogf[exp]]
OGF[binaryStringsNo00s]

$$\frac{1+z}{-1+z+z^2}$$

CoefficientList[Series[OGF[binaryStringsNo00s], {z, 0, 10}], z]
{1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144}

gen[ε, n_] := {};
gen[a_Symbol, n_] := If[n == 1, a, {}];
gen[a_ + b_, n_] := Module[{An, Bn},
  An = SeriesCoefficient[ogf[a], {z, 0, n}];
  Bn = SeriesCoefficient[ogf[b], {z, 0, n}];
  If[RandomInteger[{1, An + Bn}] ≤ An,
    gen[a, n], gen[b, n]]];
genprod[a_, b_, n_] := Module[{A, B, AB, An, x, i},
  A = PadRight[CoefficientList[Series[ogf[a], {z, 0, n}], z], n + 1];
  B = PadRight[CoefficientList[Series[ogf[b], {z, 0, n}], z], n + 1];
  AB = Accumulate[A * Reverse[B]];
  An = Last[AB];
  If[An == 0, {},
    x = RandomInteger[{1, Last@AB}];
    i = LengthWhile[AB, # < x &];
    Flatten[{gen[a, i], gen[b, n - i]}]];
gen[a_ .b_, n_] := genprod[a, b, n];
gen[a_ ^k_, n_] := genprod[a^Floor[k/2], a^Ceiling[k/2], n]

```

```

gen[a_*, n_] := If[n == 0, {}, Module[{A, AA, total, x, i},
  A = Table[SeriesCoefficient[ogf[a^k], {z, 0, n}], {k, 0, n}];
  AA = Accumulate[A];
  total = Last[AA];
  If[total == 0, {},
    x = RandomInteger[{1, total}];
    i = LengthWhile[AA, # < x &];
    gen[a^i, n]]]]

gen[a_+, n_] := genprod[a, a*, n];

GEN[expr_, n_] := Block[{$RecursionLimit = 10 000}, gen[expr, n]]

RandomBinaryStringWithNo00s[n_] := GEN[binaryStringsNo00s, n] /. {z0 -> 0, z1 -> 1}

testString = RandomBinaryStringWithNo00s[1000];

Length[testString]

1000

IntegerString[FromDigits[#, 2], 2] & /@ Split[testString]

{0, 111111, 0, 1111, 0, 11, 0, 1, 0, 1, 0, 1, 0, 1, 0, 11, 0, 11, 0, 111, 0, 11111111, 0, 11, 0, 11, 0, 11, 0, 1,
0, 1, 0, 11111, 0, 1, 0, 1111, 0, 1, 0, 1, 0, 1, 0, 111, 0, 1, 0, 111, 0, 1, 0, 11, 0, 1, 0, 11, 0, 1111,
0, 111, 0, 1, 0, 11, 0, 111, 0, 11, 0, 111111, 0, 11, 0, 1, 0, 111, 0, 1111, 0, 111111, 0, 1111111,
0, 11111, 0, 1, 0, 1, 0, 1, 0, 1, 0, 11, 0, 1, 0, 11, 0, 11, 0, 111, 0, 1111, 0, 1, 0, 111, 0, 111, 0, 1, 0,
1111, 0, 1, 0, 1, 0, 1, 0, 1, 0, 11111111111111111111111111111111, 0, 1, 0, 11111, 0, 1, 0, 111, 0,
1, 0, 1111, 0, 111, 0, 1, 0, 1111, 0, 1111, 0, 1, 0, 1111, 0, 11111111, 0, 1, 0, 111, 0, 111, 0, 1, 0,
11111111111111111111, 0, 1, 0, 1111, 0, 1, 0, 11, 0, 1, 0, 111111, 0, 1, 0, 11111, 0, 1, 0, 11, 0, 11,
0, 1, 0, 111, 0, 11111111, 0, 1, 0, 11, 0, 11, 0, 1111, 0, 1, 0, 11, 0, 111111, 0, 11111, 0, 11, 0,
11111, 0, 11111, 0, 1111, 0, 1, 0, 11, 0, 11, 0, 11, 0, 11, 0, 1, 0, 1, 0, 111, 0, 11111, 0, 1, 0, 1, 0, 11, 0,
11111, 0, 11, 0, 1, 0, 1, 0, 1111, 0, 1, 0, 1, 0, 1, 0, 1, 0, 111, 0, 1, 0, 11, 0, 1111, 0, 1, 0, 1, 0, 11, 0,
1, 0, 1111, 0, 1, 0, 1111, 0, 11, 0, 11, 0, 11, 0, 11, 0, 1, 0, 111, 0, 1, 0, 1, 0, 1, 0, 1111111, 0,
1, 0, 1111, 0, 1, 0, 1111, 0, 11, 0, 11, 0, 11, 0, 1, 0, 111, 0, 1, 0, 1, 0, 1, 0, 1111111, 0, 1, 0, 1,
0, 11111, 0, 111, 0, 11, 0, 1111, 0, 1, 0, 1, 0, 111, 0, 111111111111, 0, 1, 0, 11, 0, 1, 0, 1, 0, 1, 0,
1, 0, 11111, 0, 11, 0, 1111, 0, 1, 0, 11, 0, 11, 0, 11, 0, 111111, 0, 1, 0, 111, 0, 11, 0, 1, 0, 1, 0, 11111,
0, 11, 0, 1111, 0, 11, 0, 111, 0, 111111, 0, 1, 0, 111, 0, 111, 0, 11, 0, 1, 0, 1, 0, 1, 0, 11111, 0, 1,
0, 1, 0, 111, 0, 1, 0, 11, 0, 11, 0, 111, 0, 111111, 0, 1, 0, 111, 0, 11, 0, 1, 0, 1, 0, 1, 0, 11111, 0, 1,
0, 1111, 0, 11, 0, 111, 0, 111111, 0, 1, 0, 1, 0, 111, 0, 1111, 0, 11, 0, 1, 0, 11, 0, 1111, 0, 111,
0, 1111, 0, 1, 0, 11, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1111111111111111, 0, 1111111111, 0, 1111, 0, 111}

Average1RunLength[w_] := Mean[Map[Length, Select[Split[w], First[#] == 1 &]]]

Table[Mean[Table[Average1RunLength[RandomBinaryStringWithNo00s[k]], {1000}]],
{k, 10, 50, 10}]

{ 51 401 / 20 000 , 1 334 191 / 504 000 , 938 570 807 / 360 360 000 , 3 205 100 443 / 1 225 224 000 , 27 722 294 419 / 10 581 480 000 }

N[%]

{2.57005, 2.6472, 2.60454, 2.61593, 2.61989}

```