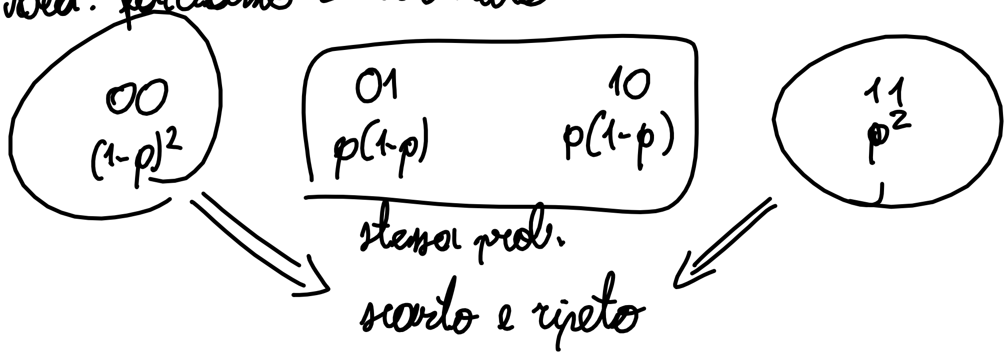
EXERCISE 1 Assume that you are given a randomized primitive BIAS(), returning 1 with probability p, and 0 with probability $1-p$, independently at each call. Assume that p is not known. Design an algorithm UNBIAS() that calls BIAS() repeatedly and returns 0/1 with probability 1/2 (clearly, UNBIAS cannot use RANDOM({0,1})). Analyze the number of calls to BIAS() needed as a function of the unknown parameter p

Idea: facciamo 2 chiamate




```
UNBIAS():
  repeat:
    a ← BIAS(); b ← BIAS();
  until a ≠ b;
  return a;
```

$\Pr[a \neq b \text{ in one it.}] = 2p(1-p) = q$

$Z = \#\text{ it.} \sim \mathrm{Geom}(q) \Rightarrow \mathbb{E}[Z] = \frac{1}{q} = \frac{1}{2p(1-p)} \Rightarrow \to 0$ per $p \to 0,\ p \to 1$

$\mathbb{E}[\#\text{ calls di BIAS}] = \mathbb{E}[2Z] = \frac{1}{p(1-p)}$

$\Pr[\text{UNBIAS}()=0]=\frac{1}{2}$

usiamo law of total prob.:

$\mathcal{F}=\{F_i : i \geq 1\}$, $\bigcup_{i=1}^{\infty} \mathcal{F}_i = \Omega$ (prob. space), $F_i \cap F_j = \emptyset \ \forall i \neq j$

$$\Pr[B] = \Pr[B \cap \Omega] = \Pr\left[B \cap \bigcup_{i=1}^{\infty} \mathcal{F}_i\right] = \Pr\left[\bigcup_{i=1}^{\infty} (B \cap \mathcal{F}_i)\right] = \sum_{i=1}^{\infty} \Pr[B \cap \mathcal{F}_i] =$$

$$= \sum_{i=1}^{\infty} \Pr[B \mid \mathcal{F}_i] \Pr[\mathcal{F}_i]$$

$$\Pr[U()=0] = \sum_{i=1}^{\infty} \Pr[U()=0 \mid \mathcal{F}_i] \Pr[\mathcal{F}_i]$$

$\mathcal{F}_i$ = "U termina dopo it. i" $\Rightarrow$ quando $\mathcal{F}_i$ succede: $\Pr[U() \to 0 \mid \mathcal{F}_i] = 1/2$

$$\Pr[U()=0] = \frac{1}{2} \sum_{i=1}^{\infty} \Pr[\mathcal{F}_i] = \frac{1}{2} \ \blacksquare$$

ADDITIONAL EXERCISES:

1. Implement RANDOM($\{0,1,2\}$) using BIAS()
2. Implement RANDOM($\{0,\dots,n-1\}$) using RANDOM($\{0,1\}$) (n ARBITRARY)

EXERCISE 2 (Partial coupon collecting)

Given a constant $c > 1$ determine an upper bound $m_c(n)$ to the number of calls to RANDOM($\{1, \dots, n\}$) so that the expected number of distinct values returned is at least $\frac{n}{c}$