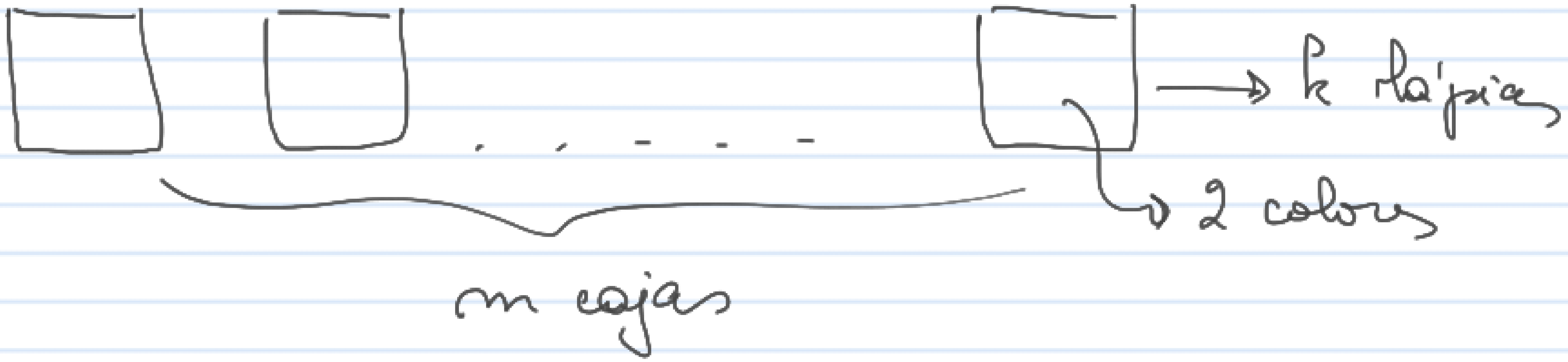
Método del alias :

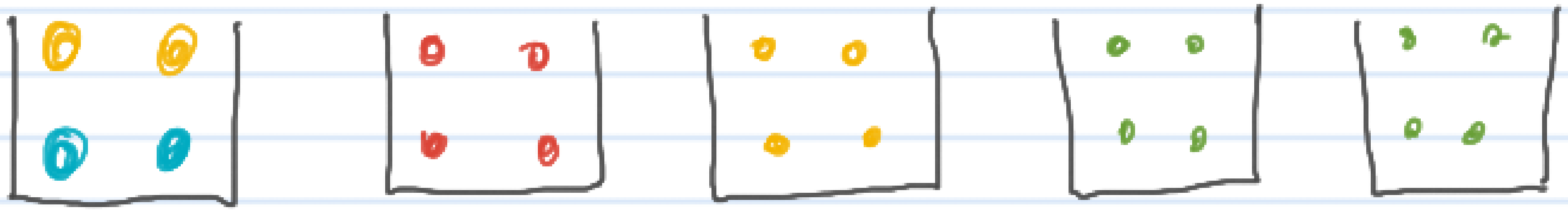
$n = m \cdot k$;   n lápices de k colores diferentes



20 lápices;   4 colores;   amarillo, verde, celeste, rojo

A : 6   V : 8   C : 2   R : 4

A : 4   C : 0   R : 0



$P(X_1 = C) = 1/2$ $P(X_1 = A) = 1/2$

X: una variable discreta que toma finitos valores.

$X \in \{1, 2, 3, 4\}$ $\quad p_1 = 0.1 \quad p_2 = 0.2 \quad p_3 = 0.3 \quad p_4 = 0.4$

n = número de valores que toma la variable

$n = 4 \qquad n - 1 = 3$

Existe un $j \mid p_j < \frac{1}{n-1}$ $\quad \left(p_j < \frac{1}{3}\right)$

De lo contrario, $\sum p_j \geq \sum \frac{1}{n-1} = \frac{n}{n-1} > 1$

Para ese j; existe un $i \neq j \mid p_i + p_j \geq 1/_{n-1}$

$$1 - p_j = \sum_{i \neq j} p_i < (n-1)\left(\frac{1}{n-1} - p_j\right) = 1 - (n-1)\, p_j$$

$(n-1)\, p_j < p_j \implies$ es cierto sólo si $n = 1$.

El método más fácil.

1) Multiplicamos todas las probabilidades por $(n-1) = 3$

i	1	2	3	4
$(n-1)\cdot p_i$	0.3	0.6	0.9	1.2

2) Buscamos $(n-1)p_j < 1$; $p_j = 0.3$

3) Buscamos $(n-1)\cdot p_i \mid (n-1)p_j + (n-1)p_i > 1$ $\quad p_i = 1.2$

4) $$X_1 = \begin{cases} 1 & \text{con } p = 0.3 \\ 4 & \text{con } p = 0.7 \end{cases}$$

Repetimos

1	2	3	4
0	0.6	0.9	0.5

Suma 2

$2 \cdot p_i$ $\qquad$ $2 \cdot p_j$

$$X_2 = \begin{cases} 4 & p = 0.5 \\ 2 & p = 0.5 \end{cases}$$

Repetimos

1	2	3	4
0	0.1	0.9	0

suma 1

$$X_3 = \begin{cases} 2 & p = 0.1 \\ 3 & p = 0.9 \end{cases}$$

Estos fueron los pasos para construir los $(n-1)$ Bernoullis.

$$X_1 , X_2 , X_3$$

El algoritmo :

```
def AliasX( ):
    I = int(random() * 3) + 1
    return X_I
```

$$P(X=2) = \frac{1}{3} P(X_1 = 2) + \frac{1}{3} P(X_2 = 2) + \frac{1}{3} P(X_3 = 2)$$

$$= \frac{1}{3} \cdot 0 + \frac{1}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{10}$$

$$= \frac{1}{3} \times \frac{6}{10} = 0{,}2$$

Generación de variables continuas

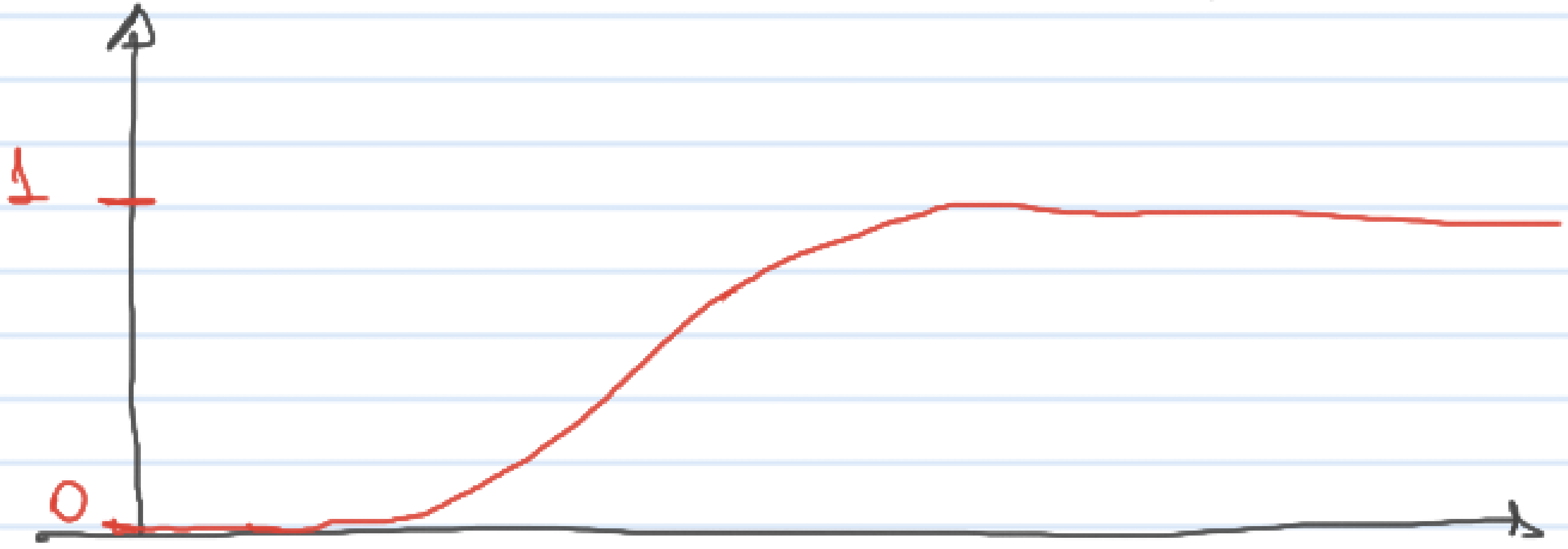
Método
- de la transformada inversa
- de aceptación y rechazo.

Método de la transformada inversa

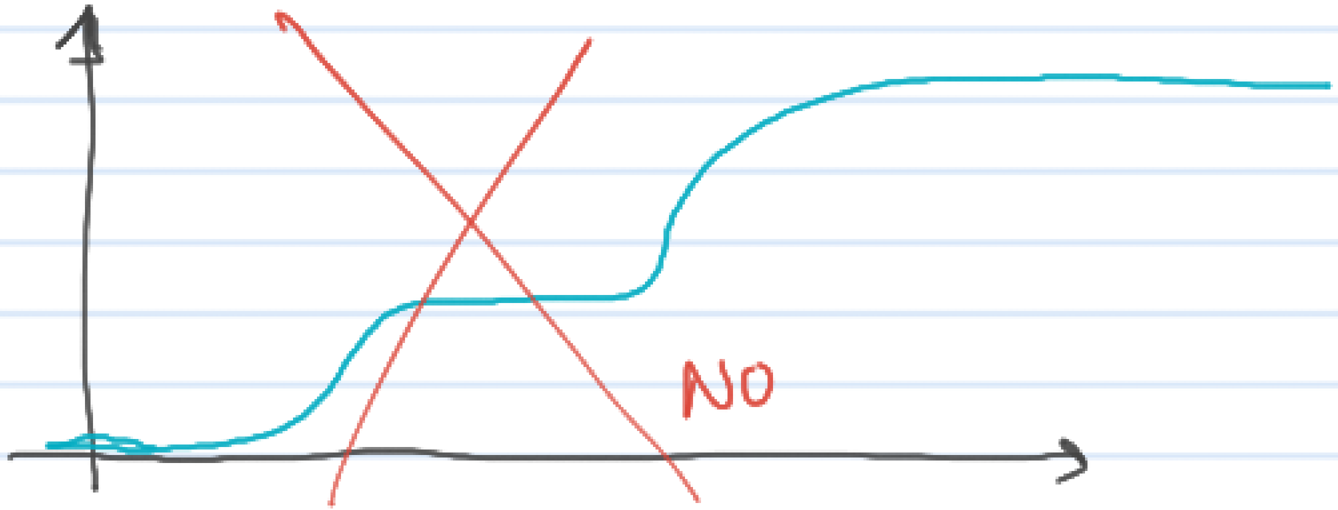
X: una v.a. absolutamente continua,

$F(x) = P(X \leq x)$: distribución acumulada; función continua

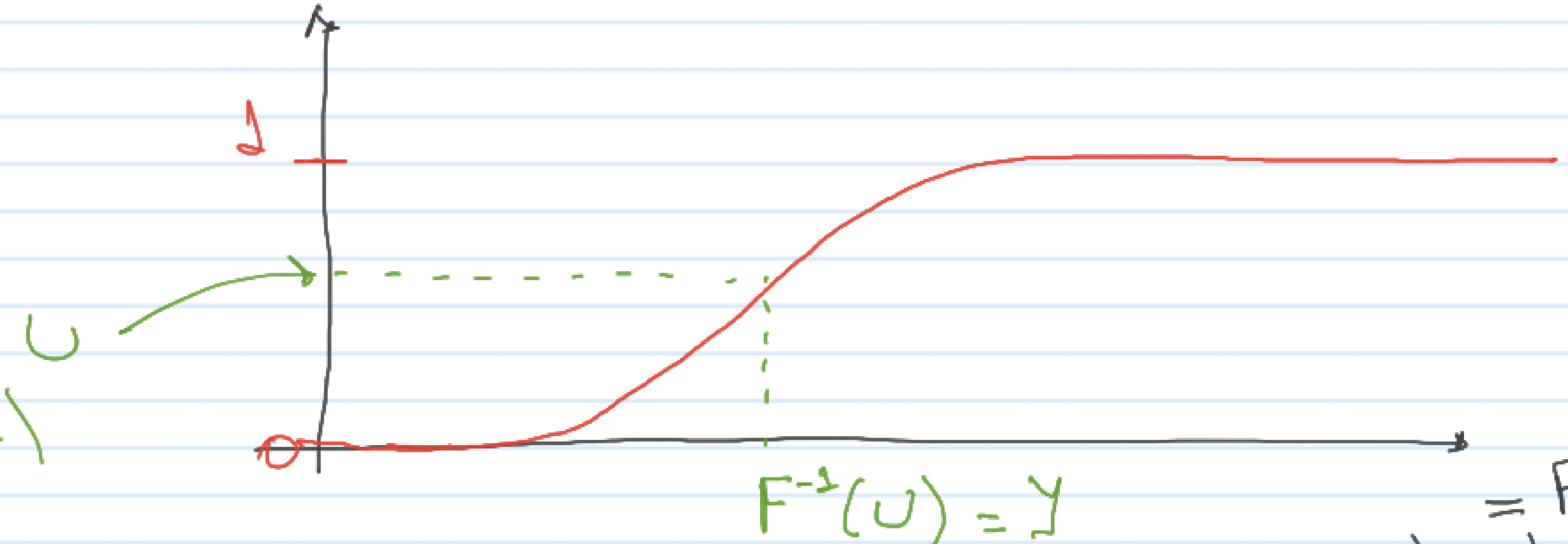
$0 \leq F(x) \leq 1$.



Vamos a considerar variables X tales que F es estrictamente creciente en el intervalo donde $F \neq 0$ y $F \neq 1$



Método.



$$P(Y \leq x) = P(F^{-1}(U) \leq x) = P(F(F^{-1}(U)) \leq F(x)) = P(U \leq F(x)) = F(x)$$

Si $G = F^{-1}(U)$; el algoritmo resulta:

```
def TInversaX():
    U = random()
    return G(U)
```

Ejemplo: Generar X con densidad

$$f(x) = \begin{cases} 1 - x/2 & \text{si } 0 \leq x \leq 2 \\ 0 & \text{en otro caso.} \end{cases}$$

1) Calculamos $F(x) = \int_{-\infty}^{x} f(t)\,dt = \begin{cases} 0 & \text{si } x \leq 0 \\ \int_0^x \left(1 - \frac{t}{2}\right) dt = x - \frac{x^2}{4} & \\ 1 & \text{si } x \geq 2 \end{cases}$

2) Calcular la inversa de $F(x)$

$$F(x) = \begin{cases} 0 & \text{si } x \leq 0 \\ x - \dfrac{x^2}{4} & 0 \leq x \leq 2 \\ 1 & \text{si } x \geq 2 \end{cases}$$

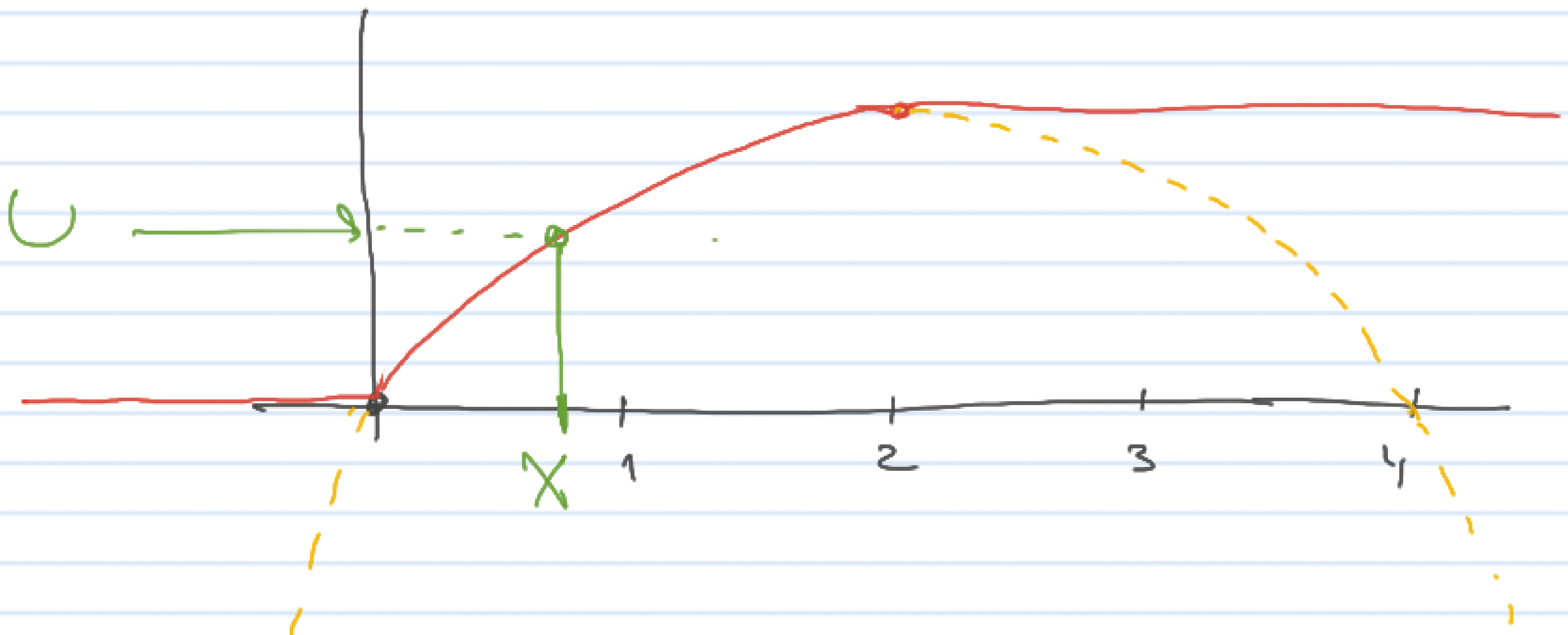
$0 < U < 1$; buscamos x tal que $x - \dfrac{x^2}{4} = U$

$$\frac{x^2}{4} - x + U = 0$$

Resolvemos la ecuación cuadrática:

$$x = \frac{1 + \sqrt{1 - U}}{1/2} \qquad x = \frac{1 - \sqrt{1-U}}{1/2}$$

$$= 2\left(1 + \sqrt{1-U}\right) \qquad = 2\left(1 - \sqrt{1-U}\right)$$



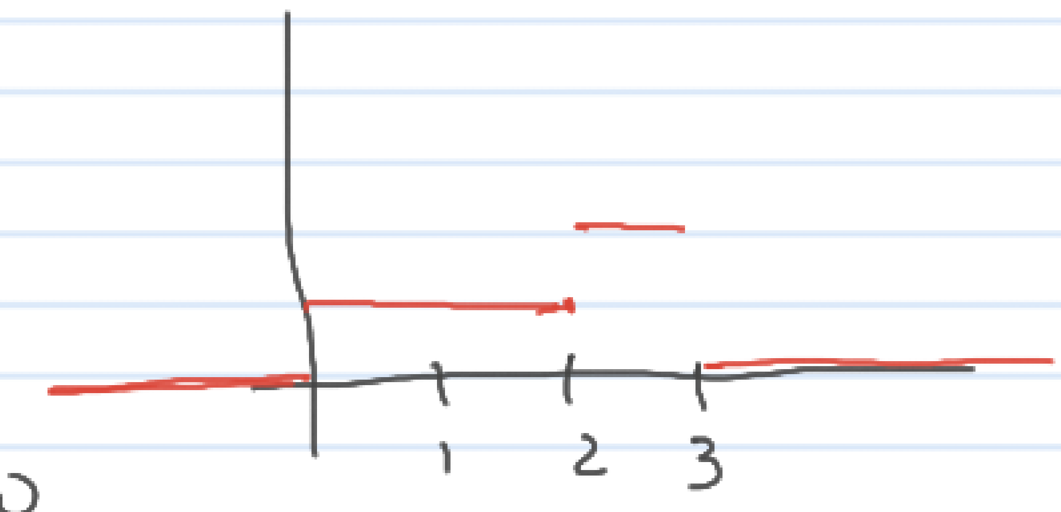
Algoritmo:

```
def TInversaX():
    U = random()
    return 2*(1 - sqrt(U))
```

↳ U y 1-U tienen la misma distribución.

Ejemplo: Generar X con densidad:

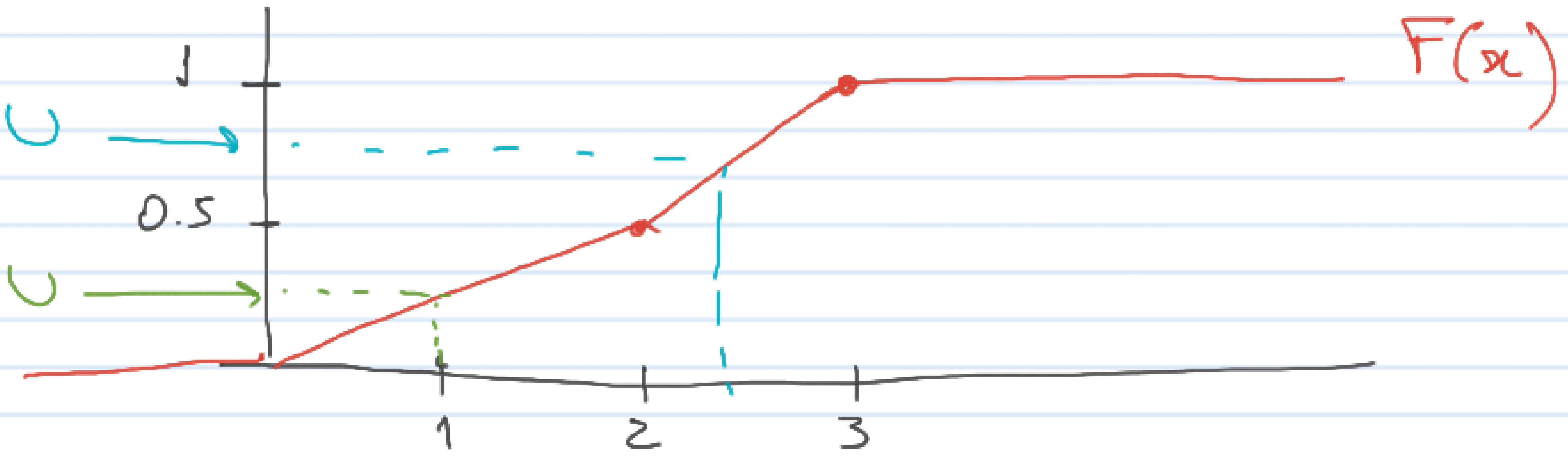
$$
f(x) = \begin{cases} 0.25 & 0 \leq x \leq 2 \\ 0.5 & 2 < x < 3 \\ 0 & \text{en otro caso} \end{cases}
$$



1) Calculamos F.

$$
F(x) = \begin{cases} 0 & \text{si } x \leq 0 \\ 0.25x & \text{si } 0 < x \leq 2 \\ 0.5 + \int_2^x 0.5\, dt = 0.5(1 + x - 2) = 0.5x - 0.5 & \\ 1 & \text{si } x \geq 3 \end{cases}
$$

Gráfico



Si $0 \leq U \leq 0.5$; $U = 0.25x$

$x = 4U$

Si $0.5 < U < 1$ $U = \frac{x-1}{2}$; $x = 2U + 1$

```
def TInversaX():
    U = random()
    if U <= 0.5:
        return 4*U
    return 2*U + 1
```

Ejemplo:

$$F(x) = \begin{cases} x^2 & ; \quad 0 \le x \le 1; \\ 0 & \text{en otro caso.} \end{cases}$$

En este caso $U = x^2 \implies x = \text{sqrt}(U)$

Otra forma de generar X con distribución F es pensar

$$F(x) = x \cdot x = F_1(x)\, F_2(x)$$

F_i: acumulada de $U_i \sim \mathcal{U}(0,1)$.

$$\begin{aligned} F_1(x) \cdot F_2(x) &= P(U_1 \le x) \cdot P(U_2 \le x) \quad ; \text{ si } U_1 \text{ y } U_2 \text{ son independientes} \\ &= P(U_1 \le x \text{ y } U_2 \le x) \\ &= P(\text{máx}\{U_1, U_2\} \le x) \end{aligned}$$

Distribución exponencial $\quad X \sim \mathcal{E}(\lambda)$

Si $\lambda = 1$; $\quad F(x) = \int_0^x e^{-t}\,dt = 1 - e^{-x}$

Para λ en general: $\quad F_\lambda(x) = 1 - e^{-\lambda x}$

Para generar $X \sim \mathcal{E}(1)$; $\quad F(x) = 1 - e^{-x} = U$

$x = ? =$

$$
\left.
\begin{aligned}
1 - e^{-x} &= U \\
e^{-x} &= 1 - U \\
-x &= \ln(1-U) \\
x &= -\ln(1-U)
\end{aligned}
\right\}
$$

Algoritmo:

```
def exp(L=1):
    return -log(1-random())
```

Si $X \sim \mathcal{E}(\lambda)$; podemos pensar $X = \frac{1}{\lambda} Y$; donde $Y \sim \mathcal{E}(1)$

ó también:

$$F_\lambda(x) = 1 - e^{-\lambda x} = U$$

$$e^{-\lambda x} = 1 - U$$

$$-\lambda x = \ln(1-U)$$

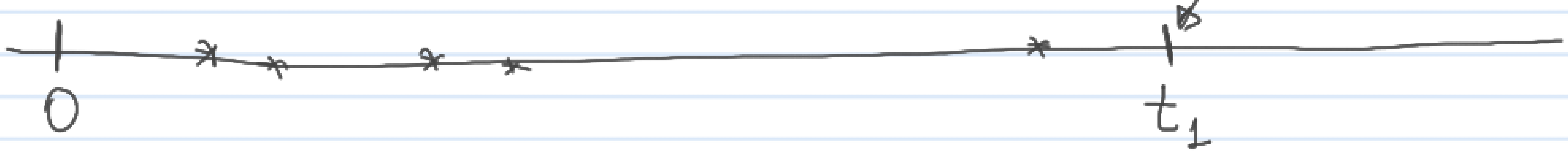
$$x = -\frac{1}{\lambda} \ln(1-U)$$

Algoritmo:

```
def exp(L):
    return -log(1-random())/L
```

Distribución de Poisson

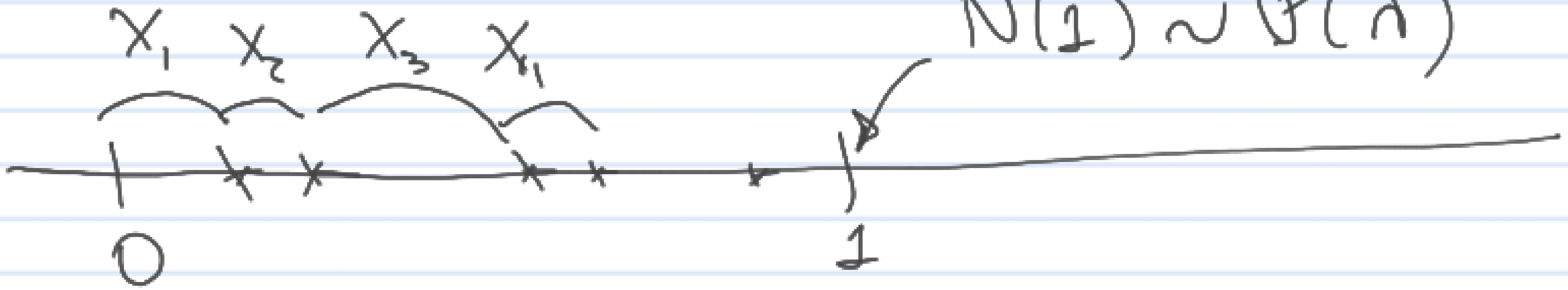
$N(t_1) \sim \mathcal{P}(\lambda \cdot t_1)$



donde $N(\cdot)$ es un proceso de Poisson con intensidad λ.

Si $t = 1$,

$N(1) \sim \mathcal{P}(\lambda)$



$X_i \sim \mathcal{E}(\lambda)$

Una estrategia es generar $X_1, X_2, \dots, X_n$ hasta que

$$X_1 + X_2 + \cdots + X_n \leq 1 \quad \text{y} \quad X_1 + X_2 + \cdots + X_n + X_{n+1} > 1$$

n : es un valor de $X \sim \mathcal{P}(\lambda)$

$$X = \text{máx} \{ n \mid X_1 + X_2 + \cdots + X_n \leq 1 \} =$$

$X_i \sim -\frac{1}{\lambda} \ln(1 - U_i)$ $\quad 1 \leq i \leq n$; todos independientes

$$= \text{máx} \left\{ n \mid -\frac{1}{\lambda} \left(\ln(1-U_1) + \ln(1-U_2) + \cdots + \ln(1-U_n) \right) \leq 1 \right\}$$

$$= \text{máx} \left\{ n \mid \ln\left((1-U_1)(1-U_2) \cdots (1-U_n) \right) \geq -\lambda \right\}$$

$$= \text{máx} \left\{ n \mid (1-U_1)(1-U_2) \cdots (1-U_n) \geq e^{-\lambda} \right\}$$

$$= \text{mín} \left\{ n \mid (1-U_1) \cdots (1-U_n) < e^{-\lambda} \right\} - 1.$$