$X \qquad S_X \qquad p$

<u>Ex</u> Bernoulli(q)

$S_X = \{0, 1\}$

$p(0) = 1-q$

$p(1) = q$

① $U \sim \text{Unif}(0,1)$

② If $U \leq 1-q$

$\Longrightarrow$ return 0

Else

$\Longrightarrow$ return 1

<u>Inverse Transform</u>

$X, \quad S_X = \{x_1 < x_2 < x_3 < \dots\}$

$p(x_i) = p_i$

① $U \sim \text{Unif}(0,1)$

② If $0 < U \leq p_1 \Longrightarrow x_1$

Elif $p_1 < U \leq p_1 + p_2 \Longrightarrow x_2$

Elif $p_1 + p_2 < U \leq p_1 + p_2 + p_3 \Longrightarrow x_3$

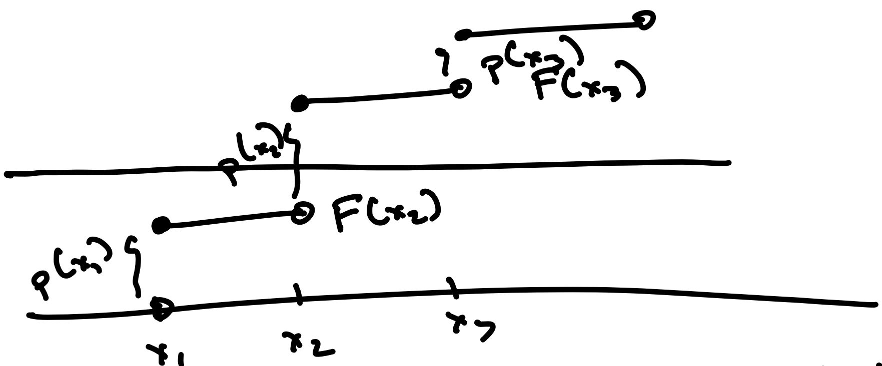
$\vdots$

Elif $\underbrace{\sum_{i=1}^{j-1} p_i}_{P(X \leq x_{j-1})} < U \leq \underbrace{\sum_{i=1}^{j} p_i}_{P(X \leq x_j)} \Longrightarrow x_j$

$\sum_{i=1}^{j-1} p_i = F(x_{j-1}) = F_{j-1}$

$\sum_{i=1}^{j} p_i = F(x_j) = F_j$

$\vdots$



how do we quantify efficiency of this algorithm?

$\Longrightarrow$ $N = \#$ of condition checks

$E[N]$ is a way of quantify the efficiency of the algorithm

Ex $S_x = \{0, 1, 2, 3\}$

$p(0) = 0.1$

$p(1) = 0.2$

$p(2) = 0.3$

$p(3) = 0.4$

$E[N] = (1)(0.1) + (2)(0.2) + (3)(0.7)$

$= 2.6$

to optimize:

① If $U \le 0.4$
return 3

Elif $U \le 0.7$
return 2

Elif $U \le 0.9$
return 1

$E[N] = (0.4)(1) + (2)(0.2) + (0.3)(3)$

$= 2$

Else return 0

Ex Discrete Uniform(n)

$S_x = \{1, \ldots, n\}$

$p(i) = 1/n \quad i \in S_x$

$nU \sim \text{Unif}(0, n)$

$\lceil nU \rceil \Rightarrow$ discrete Uniform (n)

$nU \in (0,1] \Rightarrow \lceil nU \rceil = 1 \qquad P(\lceil nU \rceil \in (0,1])$
$= 1/n$

$nU \in (i, i+1] \Rightarrow \lceil nU \rceil = i+1 \qquad P(\lceil nU \rceil \in (i, i+1])$
$= 1/n$

Ex Geo(q)

$S_x = \{1, 2, 3, \ldots\}$

$p(i) = (1-q)^{i-1} \cdot q$

$$F(n) = \sum_{i=1}^{n} p(i) = \sum_{i=1}^{n} q(1-q)^{i-1}$$

$$= q \sum_{i=0}^{n-1} (1-q)^i$$

$$\boxed{\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}}$$

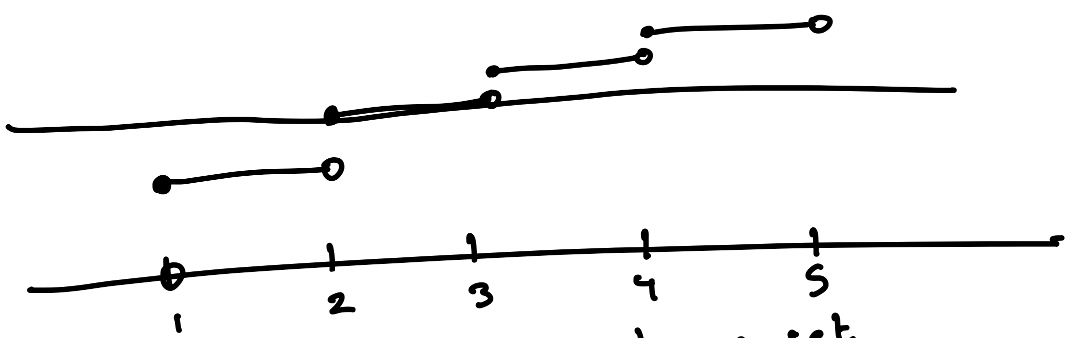
$$= q \, \frac{1-(1-q)^n}{1-(1-q)}$$

$$= 1-(1-\delta)^n$$

suppose $u = 1-(1-\delta)^n$

$$(1-\delta)^n = 1-u$$

$$n = \left\lceil \frac{\log(1-u)}{\log(1-\delta)} \right\rceil$$



If we just do a search, we set

$$E[N] = 1/\delta$$

Generate a sequence of Bernulli(δ) random variables

$$X_1, \dots, X_n \overset{iid}{\sim} \text{Bern}(\delta)$$

Method 1: $U_1, \dots, U_n \overset{iid}{\sim} \text{Unif}(0,1)$

$$X_i = \begin{cases} 0, & U_i \le (1-\delta) \\ 1, & U_i > (1-\delta) \end{cases}$$

Method 2: recall geo(q) is the 1st bernoulli in an iid sequence to be equal to 1

$N = \text{geo}(q)$ if $N \leq n$

$X_0 = X_1 = \dots = X_{N-1} = 0$

$X_N = 1$

repeat process, remaining $N-n$ bernoulli

if $N > n$

then I'm done with

$X_0 = X_1 = \dots = X_N = 0$

Generating Poisson Random Variables

$X \sim \text{Poiss}(\lambda)$

$S_x = \{0, 1, 2, \dots\}$

$p(x) = \frac{e^{-\lambda}\lambda^x}{x!}$ $E[X] = \lambda$

$Var(X) = \lambda$

$$P_i = p(i) = \frac{e^{-\lambda}\lambda^i}{i!}$$

$$= \frac{\lambda}{} \cdot \left(\frac{e^{-\lambda}\lambda^{i-1}}{} \right)$$

$$i \quad (i-1)! \quad /$$

$$= \frac{\lambda}{i} \cdot p_{i-1} \quad]$$

we can do the inverse transform as follows:

① $U \sim Unif(0,1)$

② $i = 0$, $p = e^{-\lambda}$, $F = p$

③ If $U \leq F$ return i

Else $i = i+1$

$$p = \frac{\lambda}{i} \cdot p \quad]$$

$F = F + p$

repeat step ③

$E[N] = E[X+1]$ b/c $N = X+1$

$= \lambda + 1$

$\Longrightarrow$ this is going to be slow when λ is big

$$p(x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

informally to speed this up, we want to search starting from the vals of x with largest pmf

1. $I = \text{Int}(\lambda)$
2. calculate $F(I)$, $p(I)$

③ $U \sim \text{Unif}(0,1)$

$F = F(I)$

$p = p(I)$

$i = I$

④ If $U \le F(I)$

④a let $F = F - p$

if $U \geq F$

return i

Else

$i = i - 1$

$p = \frac{i}{\lambda} \cdot p$

repeat ④a

(5) Inverse transform starting from

30

expected # of comparisons

$X - \lambda \sim N(0, \lambda)$

$$\approx 1 + E[|X - \lambda|]$$

$$\approx 1 + \sqrt{\lambda}\, E\left[\frac{|X-\lambda|}{\sqrt{\lambda}}\right]$$

$$= 1 + \sqrt{\lambda}\, E[|Z|] \qquad |Z| \sim N(0,1)$$

$$= 1 + 0.8\sqrt{\lambda}$$