Randomized Algorithm

Probability

1. (Discrete) random variable
2. Expectation

Q: "Coins ⇒ p heads, (1-p) tails. How many flips → See first Head?"

"Waiting Time Problem"

※ Showing that for $p \in (0,1)$

$\sum_{i=1}^{\infty} i(1-p)^{i-1} = \frac{1}{p^2}$

Set $x = 1-p$. $x \in (0,1)$ let $k = i-1$

$S = \sum_{i=1}^{\infty} i(1-p)^{i-1} = \sum_{i=1}^{\infty} i x^{i-1} = \sum_{k=0}^{\infty} (k+1)x^k = 1 + 2x + 3x^2 + 4x^3 + \cdots$

$\Rightarrow S = 1 + 2x + 3x^2 + 4x^3 \ldots$

$xS = \quad x + 2x^2 + 3x^3 + \cdots$

$\Rightarrow (1-x)S = \underbrace{1 + x + x^2 + x^3 + \cdots = \frac{1}{1-x}}_{\text{Geometric Series!}}$

$\Rightarrow S = \frac{1}{(1-x)^2} = \frac{1}{p^2}$

Let Waiting Time be random variable T.

$E(T) = \sum_{i=1}^{\infty} i \cdot P(T) = \sum_{i=1}^{\infty} i \cdot \underbrace{(1-p)^{i-1}}_{i-1 \text{ times tail}} \cdot \underbrace{p}_{\text{last time head}}$

$= p \cdot \frac{1}{p^2} = \frac{1}{p}$

$E(X+Y) = E(X) + E(Y)$

$E(X \cdot Y) = E(X)E(Y)$ ONLY When X, Y are independent

Coupon Collector

n difference type of coupon

how many boxes needs to be open to collect all kinds of coupon

Stage i → having i coupon. want to have $i+1$ coupon.

probability: $\frac{n-i}{n}$

Let X_i be number of boxes to open to get the extra coupon

$E(X_i) = \frac{1}{\frac{n-i}{n}} = \frac{n}{n-i}$

(See waiting time problem)

Let X be the number of boxes to open to get all coupons.

$E(X) = \sum_{i=1}^{n} E(X_i) = \sum_{i=0}^{n-1} \frac{n}{n-i}$

$= 1 + \frac{n}{n-1} + \frac{n}{n-2} \cdots + \frac{n}{1}$

$= n \cdot \underbrace{\sum_{i=0}^{n-1} \frac{1}{i}}_{\text{"}\Theta(\log n)\text{"}} = \Theta(n \log n)$

Guessing with no memory.

$E(X_i) = \frac{1}{n}$ (n cards)

$E(X) = \sum_{i=1}^{n} E(X_i) = 1$

Guessing with memory

$E(X_i) = \frac{1}{n-i+1}$

$E(X) = \sum_{i=1}^{n} E(X_i) = \frac{1}{n} + \frac{1}{n-1} + \cdots + 1 = \Theta(\log n)$

Hiring Problem

```
Hire-Assistant(n):          random permutate all n candidates.
  best ← 0
  for i ← 1 to n:
      interview candidate i
      if candidate i is better than best then
        ☆ hire candidate i
          best ← i
```

How many people are hired in Worst Case:

⇒ n (all people)

How to avoid randomness of inputs?

add random in algorithm

Expected number:

$E(X_i) = P(X_i \text{ is better than best}) = \frac{1}{i}$

$\therefore E(X) = \sum_{i=1}^{n} E(X_i) = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = \Theta(\log n)$

Birthday Paradox

k people. How large should be k for us to expect there are 2 people have same birthday?

Let n be # days in a year.

$X_{ij} = 1$ if i, j have same birthday ⇒ $E(X_{ij}) = P(i, j \text{ have same birthday}) = \frac{1}{n}$

$E(X) = \sum_{1 \le i < j \le k} E(X_{ij}) = \binom{k}{2} \cdot \frac{1}{n} = \frac{k(k-1)}{2n}$

We want $E(X) \ge 1$ then $k(k-1) \ge 2n \Rightarrow k \ge \sqrt{2n} + 1 \approx 28$

Generating Random Permutation

⇔ generating each permutation with probability $1/n!$ (n items)

```
Random-Permute(A):
  n ← A.length
  for i ← 1 to n:
      swap A[i] with A[Random(1, i)]
```

Why this algorithm ensures each one's probability is $1/n!$?

Prove By induction:

Base case: $n = 1$, $P = 1$, trivially correct.

Assume: $A[1..n-1]$ have been randomly permuted correctly (each one's probability: $\frac{1}{(n-1)!}$)

Consider: $A[1..n]$, first $n-1$ iteration ⇒ $A[1..n-1]$ have been randomly permuted correctly

a_n' → last element in A before permutation

↓ swap a_n' with random(1, n) $\frac{1}{n}$.

a_n

↳ probability of $(a_1 \ldots a_n)$ is $\frac{1}{(n-1)!} \cdot \frac{1}{n} = \frac{1}{n!}$ ⇒ random permutation

Generating: $(a_1 \ldots a_j \ldots a_{n-1}) \rightarrow$ "$\frac{1}{(n-1)!}$"

$a_n \overset{\text{swap}}{\leftrightarrows} a_j \rightarrow \frac{1}{n}$ since n possible a_j

$(a_1 \ldots a_n \ldots a_{n-1}\ a_j) \longrightarrow \frac{1}{(n-1)! \cdot n}$