Example 3: The Hat-Check Problem

Exercise 5.2-4:
Each of n customers gives a hat to a hat-check person at a restaurant.
The hat-check person gives the hats back to the customers in a random order.
What is the expected number of customers who get back their own hat?

Solution:

- Let X be a random variable equal to the number of customers who get their own hat.
- For $i \in \{1, 2, \ldots, n\}$, define an indicator random variable $X_i = I\{\text{customer } i \text{ gets his/her own hat}\}$.
- Then $X = \sum_{i=1}^{n} X_i$.
- $Pr\{\text{customer } i \text{ gets his/her own hat}\} = 1/n$
- According to Lemma 5.1, $E[X_i] = 1/n$.
- By linearity of expectation:
  $E[X] = E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} 1/n = \boxed{1}$

Monte Carlo algorithm, example

The Matrix Product Verification Problem

Input: Three matrices A, B, C of size $(n \times n)$.
Output: "YES" if $A\,B = C$; "NO" otherwise.

Compute $A\,B$ from scratch and compare it to $C \Rightarrow \boxed{O(n^{2.38})}$ time by Lecture 2.

A faster method (Monte Carlo algorithm), known as Freivalds' algorithm:

> Pick a random vector $r = (r_1, r_2, \ldots, r_n)$ where each $r_i \in \{0, 1\}$.
> **if** $A\,(B\,r) = C\,r$ **then** output "YES"
> **else** output "NO"

- Runs in $O(n^2)$ time.
- If $A\,B = C$ then the algorithm always correctly outputs "YES".
- If $A\,B \neq C$ then the algorithm may incorrectly output "YES".
  We will see in Tutorial 4 that $Pr\{\text{error}\} \leq \frac{1}{2}$.

Repeat t times, and output "YES" if and only if the answer was always "YES".
$\Rightarrow \boxed{O(tn^2)}$ time and $Pr\{\text{error}\} \leq \boxed{2^{-t}}$.

Proof: (Freivalds' algorithm)

$AB \neq C \Rightarrow AB - C \neq 0$

There exist indices i, j such that the entry on row i and column j of the matrix $(AB - C)$ is not equal to 0.

Let $[d_1, d_2, ..., d_n]$ be row i of $(AB - C)$. By definition, $d_j \neq 0$.

$Pr\{error\} = Pr\{(AB - C)r = 0 \,|\, AB \neq C\}$
$\leq Pr\{d_1 r_1 + d_2 r_2 + ... + d_n r_n = 0 \,|\, AB \neq C\}$

reason: $d_1 r_1 + d_2 r_2 + ... + d_n r_n$ is one element of the vector $(AB - C)r$.

Therefore, the probability that all elements in the vector $(AB - C)r$ are zero is less than or equal to the probability that $d_1 r_1 + d_2 r_2 + \ldots + d_n r_n = 0$.

$= Pr\{r_j = -\frac{1}{d_j} \sum_{k \neq j} d_k r_k \,|\, AB \neq C\} \leq \frac{1}{2}$

reason: r_j is selected uniformly at random from $\{0, 1\}$, independently of A, B, and C.
Therefore, the probability that r_j is exactly equal to some specified value is at most $\frac{1}{2}$.

Questions: Randomized Algorithms

(1)

You are given a function *Random()* such that:
- it returns 0 with probability 1/2,
- it returns 1 with probability 1/2.

By calling the function *Random()*, write an algorithm such that:
- it is a Las-Vegas algorithm,
- it takes a positive integer n as input,
- it returns a random integer between 1 and n (with the same probability).

Analyze the expected time complexity of your algorithm.

(2)

You are given a function *Random-Bias()* such that:
- it returns 0 with probability p,
- it returns 1 with probability $1–p$.

where p is a constant and $0 < p < 1$. However, you do not know the value of p.
By calling the function *Random-Bias()*, write an algorithm such that:
- it is a Las-Vegas algorithm,
- it returns 0 with probability 1/2; it returns 1 with probability 1/2.

Express the expected time complexity of your algorithm in terms of p.