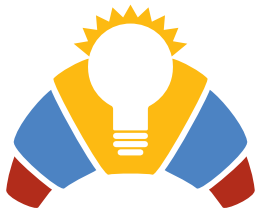
Algoritmos Randomizados

Avanzado

Matías Fluxa

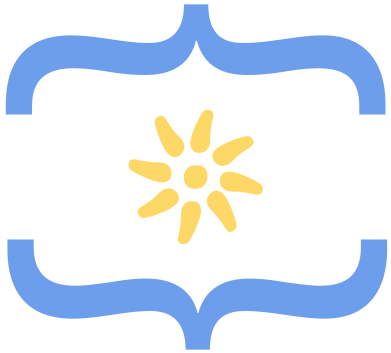
Universidad Nacional de La Plata

Training Camp 2026



Gracias Sponsors!

Organiza



Facultad de INFORMÁTICA

UNIVERSIDAD NACIONAL DE LA PLATA

Diamond Sponsor

Outline

1. Repaso rápido de probabilidad
2. Ejercicio divertido
3. Montecarlo
4. Las Vegas
5. Estructuras de datos aleatorias
6. Conclusiones y cierre

Outline

1. Repaso rápido de probabilidad
2. Ejercicio divertido
3. Montecarlo
4. Las Vegas
5. Estructuras de datos aleatorias
6. Conclusiones y cierre

Definición de probabilidad

Definición

La probabilidad de un evento A es un número entre 0 y 1 que representa la relación entre la cantidad de eventos de éxito y la cantidad de eventos posibles. Se denota como $P(A)$.

Definición de probabilidad

Definición

La probabilidad de un evento A es un número entre 0 y 1 que representa la relación entre la cantidad de eventos de éxito y la cantidad de eventos posibles. Se denota como $P(A)$.

Por ejemplo, si lanzamos un dado balanceado, la probabilidad de obtener un número par es:

$$P(\text{número par}) = \frac{\#\ \text{Números pares}}{\#\ \text{Números Totales}} = \frac{3}{6} = \frac{1}{2}$$

Esperanza

Esperanza

La esperanza de una variable aleatoria X es el valor promedio que se espera obtener si se repite el experimento un gran número de veces. Se denota como $\mathbb{E}[X]$.

Esperanza

Esperanza

La esperanza de una variable aleatoria X es el valor promedio que se espera obtener si se repite el experimento un gran número de veces. Se denota como $\mathbb{E}[X]$.

La esperanza se calcula como:

$$\mathbb{E}[X] = \sum_{x} x \cdot P(X = x)$$

Esperanza

Esperanza

La esperanza de una variable aleatoria X es el valor promedio que se espera obtener si se repite el experimento un gran número de veces. Se denota como $\mathbb{E}[X]$.

La esperanza se calcula como:

$$\mathbb{E}[X] = \sum_{x} x \cdot P(X = x)$$

Por ejemplo, si lanzamos un dado balanceado, la esperanza del número que obtenemos es:

$$\mathbb{E}[X] = \sum_{i=1}^{6} i \cdot P(X = i) = \sum_{i=1}^{6} i \cdot \frac{1}{6} = \frac{1+2+3+4+5+6}{6} = \frac{21}{6} = 3.5$$

Varianza y Desviación Estándar

Varianza

La varianza de una variable aleatoria X mide la dispersión de los valores posibles de X respecto a su esperanza. Se denota como $\text{Var}(X)$.

La varianza se calcula como:

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

Varianza y Desviación Estándar

Varianza

La varianza de una variable aleatoria X mide la dispersión de los valores posibles de X respecto a su esperanza. Se denota como $\text{Var}(X)$.

La varianza se calcula como:

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

Por ejemplo, si lanzamos un dado balanceado, la varianza del número que obtenemos es:

$$\text{Var}(X) = \sum_{i=1}^{6} (i - 3.5)^2 \cdot P(X = i) = \sum_{i=1}^{6} (i - 3.5)^2 \cdot \frac{1}{6} = \frac{17.5}{6} \approx 2.9167$$

Varianza y Desviación Estándar

Varianza

La varianza de una variable aleatoria X mide la dispersión de los valores posibles de X respecto a su esperanza. Se denota como $\text{Var}(X)$.

La varianza se calcula como:

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

Por ejemplo, si lanzamos un dado balanceado, la varianza del número que obtenemos es:

$$\text{Var}(X) = \sum_{i=1}^{6} (i - 3.5)^2 \cdot P(X = i) = \sum_{i=1}^{6} (i - 3.5)^2 \cdot \frac{1}{6} = \frac{17.5}{6} \approx 2.9167$$

Desviación Estándar

Desviación estándar

La desviación estándar de una variable aleatoria X mide la dispersión de los valores posibles de X respecto a su esperanza en las mismas unidades que X. Se denota como $\text{Std}(X)$ y se calcula como:

$$\text{Std}(X) = \sqrt{\text{Var}(X)}$$

Desviación Estándar

Desviación estándar

La desviación estándar de una variable aleatoria X mide la dispersión de los valores posibles de X respecto a su esperanza en las mismas unidades que X. Se denota como $\text{Std}(X)$ y se calcula como:

$$\text{Std}(X) = \sqrt{\text{Var}(X)}$$

Por ejemplo, si lanzamos un dado balanceado, la desviación estándar del número que obtenemos es:

$$\text{Std}(X) = \sqrt{\text{Var}(X)} = \sqrt{2.9167} \approx 1.7078$$

Propiedades de la Esperanza

Linealidad de la Esperanza

Para variables aleatorias X e Y y constantes a, b:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

Propiedades de la Esperanza

Linealidad de la Esperanza

Para variables aleatorias X e Y y constantes a, b:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

Esta propiedad es **fundamental** en análisis de algoritmos aleatorios porque:

- Permite descomponer problemas complejos en partes más simples

Propiedades de la Esperanza

Linealidad de la Esperanza

Para variables aleatorias X e Y y constantes a, b:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

Esta propiedad es **fundamental** en análisis de algoritmos aleatorios porque:

- Permite descomponer problemas complejos en partes más simples
- No requiere que X e Y sean independientes

Propiedades de la Esperanza

Linealidad de la Esperanza

Para variables aleatorias X e Y y constantes a, b:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

Esta propiedad es **fundamental** en análisis de algoritmos aleatorios porque:

- Permite descomponer problemas complejos en partes más simples
- No requiere que X e Y sean independientes
- Simplifica el cálculo de esperanzas de sumas

Propiedades de la Esperanza

Linealidad de la Esperanza

Para variables aleatorias X e Y y constantes a, b:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

Esta propiedad es **fundamental** en análisis de algoritmos aleatorios porque:

- Permite descomponer problemas complejos en partes más simples
- No requiere que X e Y sean independientes
- Simplifica el cálculo de esperanzas de sumas

Propiedades de la Esperanza

Linealidad de la Esperanza

Para variables aleatorias X e Y y constantes a, b:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

Esta propiedad es **fundamental** en análisis de algoritmos aleatorios porque:

- Permite descomponer problemas complejos en partes más simples
- No requiere que X e Y sean independientes
- Simplifica el cálculo de esperanzas de sumas

Ejemplo: Si X_i es el número de comparaciones en la i-ésima iteración, entonces:

$$\mathbb{E}[\text{total comparaciones}] = \mathbb{E}[X_1 + X_2 + \cdots + X_n] = \sum_{i=1}^{n} \mathbb{E}[X_i]$$

Ejemplo: Moneda Desbalanceada

Lanzamos una moneda **desbalanceada** 100 veces, con probabilidad p de obtener cara.

Ejemplo: Moneda Desbalanceada

Lanzamos una moneda **desbalanceada** 100 veces, con probabilidad p de obtener cara.
Sea $X_i = 1$ si el lanzamiento i es cara, $X_i = 0$ en caso contrario.

$$\mathbb{E}[X_i] = 1 \cdot p + 0 \cdot (1 - p) = p$$

Ejemplo: Moneda Desbalanceada

Lanzamos una moneda **desbalanceada** 100 veces, con probabilidad p de obtener cara.
Sea $X_i = 1$ si el lanzamiento i es cara, $X_i = 0$ en caso contrario.

$$\mathbb{E}[X_i] = 1 \cdot p + 0 \cdot (1 - p) = p$$

Sea $X = X_1 + X_2 + \cdots + X_{100}$ el total de caras.

Ejemplo: Moneda Desbalanceada

Lanzamos una moneda **desbalanceada** 100 veces, con probabilidad p de obtener cara.
Sea $X_i = 1$ si el lanzamiento i es cara, $X_i = 0$ en caso contrario.

$$\mathbb{E}[X_i] = 1 \cdot p + 0 \cdot (1 - p) = p$$

Sea $X = X_1 + X_2 + \cdots + X_{100}$ el total de caras.
Por linealidad de la esperanza:

$$\mathbb{E}[X] = \mathbb{E}[X_1 + X_2 + \cdots + X_{100}] = \sum_{i=1}^{100} \mathbb{E}[X_i] = \sum_{i=1}^{100} p = 100p$$

Ejemplo: Moneda Desbalanceada

Lanzamos una moneda **desbalanceada** 100 veces, con probabilidad p de obtener cara.
Sea $X_i = 1$ si el lanzamiento i es cara, $X_i = 0$ en caso contrario.

$$\mathbb{E}[X_i] = 1 \cdot p + 0 \cdot (1 - p) = p$$

Sea $X = X_1 + X_2 + \cdots + X_{100}$ el total de caras.
Por linealidad de la esperanza:

$$\mathbb{E}[X] = \mathbb{E}[X_1 + X_2 + \cdots + X_{100}] = \sum_{i=1}^{100} \mathbb{E}[X_i] = \sum_{i=1}^{100} p = 100p$$

Example

Si $p = 0.7$, esperamos obtener $100 \cdot 0.7 = 70$ caras en promedio.
Sin linealidad, tendríamos que calcular todas las combinaciones binomiales directamente.

Propiedades de la Varianza

Varianza de la Suma

Para variables aleatorias independientes X e Y:

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$$

Propiedades de la Varianza

Varianza de la Suma

Para variables aleatorias independientes X e Y:

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$$

Varianza de la Escala

Para una variable aleatoria X y una constante a:

$$\text{Var}(aX) = a^2\text{Var}(X)$$

Propiedades de la Varianza

Varianza de la Suma

Para variables aleatorias independientes X e Y:

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$$

Varianza de la Escala

Para una variable aleatoria X y una constante a:

$$\text{Var}(aX) = a^2\text{Var}(X)$$

Caso General: Variables Independientes No Idénticamente Distribuidas

Sea $X_1, X_2, \ldots, X_n$ variables aleatorias **independientes** con distintas esperanzas μ_i y distintas varianzas σ_i^2. Definimos su promedio como:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Caso General: Variables Independientes No Idénticamente Distribuidas

Sea $X_1, X_2, \ldots, X_n$ variables aleatorias **independientes** con distintas esperanzas μ_i y distintas varianzas σ_i^2. Definimos su promedio como:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

La esperanza es el promedio aritmético de las esperanzas individuales:

$$\mathbb{E}[\bar{X}] = \frac{1}{n} \sum_{i=1}^{n} \mu_i = \frac{\mu_1 + \mu_2 + \cdots + \mu_n}{n}$$

Caso General: Variables Independientes No Idénticamente Distribuidas

Sea $X_1, X_2, \ldots, X_n$ variables aleatorias **independientes** con distintas esperanzas μ_i y distintas varianzas σ_i^2. Definimos su promedio como:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

La esperanza es el promedio aritmético de las esperanzas individuales:

$$\mathbb{E}[\bar{X}] = \frac{1}{n} \sum_{i=1}^{n} \mu_i = \frac{\mu_1 + \mu_2 + \cdots + \mu_n}{n}$$

La varianza es la suma de las varianzas individuales dividida por n^2:

$$\mathrm{Var}(\bar{X}) = \frac{1}{n^2} \sum_{i=1}^{n} \sigma_i^2 = \frac{\bar{\sigma}^2}{n}$$

donde $\bar{\sigma}^2$ representa la varianza promedio de los datos.

Caso General: Variables Independientes No Idénticamente Distribuidas

Sea $X_1, X_2, \ldots, X_n$ variables aleatorias **independientes** con distintas esperanzas μ_i y distintas varianzas σ_i^2. Definimos su promedio como:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

La esperanza es el promedio aritmético de las esperanzas individuales:

$$\mathbb{E}[\bar{X}] = \frac{1}{n} \sum_{i=1}^{n} \mu_i = \frac{\mu_1 + \mu_2 + \cdots + \mu_n}{n}$$

La varianza es la suma de las varianzas individuales dividida por n^2:

$$\mathrm{Var}(\bar{X}) = \frac{1}{n^2} \sum_{i=1}^{n} \sigma_i^2 = \frac{\bar{\sigma}^2}{n}$$

donde $\bar{\sigma}^2$ representa la varianza promedio de los datos.

Conclusión clave

Caso Especial: Variables Independientes e Idénticamente Distribuidas (i.i.d.)

Sea $X_1, X_2, \ldots, X_n$ variables aleatorias independientes e idénticamente distribuidas (i.i.d.) con esperanza μ y varianza σ^2. Definimos el promedio

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Caso Especial: Variables Independientes e Idénticamente Distribuidas (i.i.d.)

Sea $X_1, X_2, \ldots, X_n$ variables aleatorias independientes e idénticamente distribuidas (i.i.d.) con esperanza μ y varianza σ^2. Definimos el promedio

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Entonces

$$\mathbb{E}[\bar{X}] = \mu$$

y

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

Caso Especial: Variables Independientes e Idénticamente Distribuidas (i.i.d.)

Sea $X_1, X_2, \ldots, X_n$ variables aleatorias independientes e idénticamente distribuidas (i.i.d.) con esperanza μ y varianza σ^2. Definimos el promedio

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Entonces

$$\mathbb{E}[\bar{X}] = \mu$$

y

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

Esto significa que al promediar más muestras, la varianza del promedio disminuye, lo que nos da una estimación más precisa de la esperanza verdadera.

Outline

1. Repaso rápido de probabilidad
2. Ejercicio divertido
3. Montecarlo
4. Las Vegas
5. Estructuras de datos aleatorias
6. Conclusiones y cierre

Ejercicio divertido: Aproxima π

Problema Todos sabemos muy bien que los primeros 4 dígitos de π son 3.1415. Supongamos que estamos en una prueba de icpc y no nos acordamos los dígitos de π. ¿Cómo podemos aproximar π rápidamente?

Solución

Para resolver este problema, podemos plantear otro problema.

Solución

Para resolver este problema, podemos plantear otro problema.

Problema equivalente: Supongamos que tenemos un cuadrado de lado 2 y un círculo inscrito en él. Si lanzamos n dardos al azar, ¿cuál es la probabilidad de que caigan dentro del círculo?

Consideramos que cada dardo tiene la misma probabilidad de caer en cualquier punto del cuadrado.

Planteamiento Formal

Coordenadas

Cada dardo es una coordenada (x, y) donde:

- $x \in [0, 2]$ distribuida uniformemente
- $y \in [0, 2]$ distribuida uniformemente
- El círculo tiene centro en $(1, 1)$ y radio 1

Planteamiento Formal

Coordenadas

Cada dardo es una coordenada (x, y) donde:

- $x \in [0, 2]$ distribuida uniformemente
- $y \in [0, 2]$ distribuida uniformemente
- El círculo tiene centro en $(1, 1)$ y radio 1

Un dardo cae **dentro del círculo** si:

$$(x - 1)^2 + (y - 1)^2 \leq 1$$

Planteamiento Formal

Coordenadas

Cada dardo es una coordenada (x, y) donde:

- $x \in [0, 2]$ distribuida uniformemente
- $y \in [0, 2]$ distribuida uniformemente
- El círculo tiene centro en $(1, 1)$ y radio 1

Un dardo cae **dentro del círculo** si:

$$(x - 1)^2 + (y - 1)^2 \leq 1$$

Cálculo de probabilidad:

- Área del cuadrado: $2 \times 2 = 4$

Planteamiento Formal

Coordenadas

Cada dardo es una coordenada (x, y) donde:

- $x \in [0, 2]$ distribuida uniformemente
- $y \in [0, 2]$ distribuida uniformemente
- El círculo tiene centro en $(1, 1)$ y radio 1

Un dardo cae **dentro del círculo** si:

$$(x - 1)^2 + (y - 1)^2 \leq 1$$

Cálculo de probabilidad:

- Área del cuadrado: $2 \times 2 = 4$
- Área del círculo: $\pi \cdot 1^2 = \pi$

Planteamiento Formal

Coordenadas

Cada dardo es una coordenada (x, y) donde:

- $x \in [0, 2]$ distribuida uniformemente
- $y \in [0, 2]$ distribuida uniformemente
- El círculo tiene centro en $(1, 1)$ y radio 1

Un dardo cae **dentro del círculo** si:

$$(x - 1)^2 + (y - 1)^2 \leq 1$$

Cálculo de probabilidad:

- Área del cuadrado: $2 \times 2 = 4$
- Área del círculo: $\pi \cdot 1^2 = \pi$
- Probabilidad de caer dentro: $P = \frac{\pi}{4}$

Aproximando π

Si lanzamos n dardos y k caen dentro del círculo, entonces:

$$\frac{k}{n} \approx \frac{\pi}{4}$$

Aproximando π

Si lanzamos n dardos y k caen dentro del círculo, entonces:

$$\frac{k}{n} \approx \frac{\pi}{4}$$

Despejando π:

$$\pi \approx 4 \cdot \frac{k}{n}$$

Aproximando π

Si lanzamos n dardos y k caen dentro del círculo, entonces:

$$\frac{k}{n} \approx \frac{\pi}{4}$$

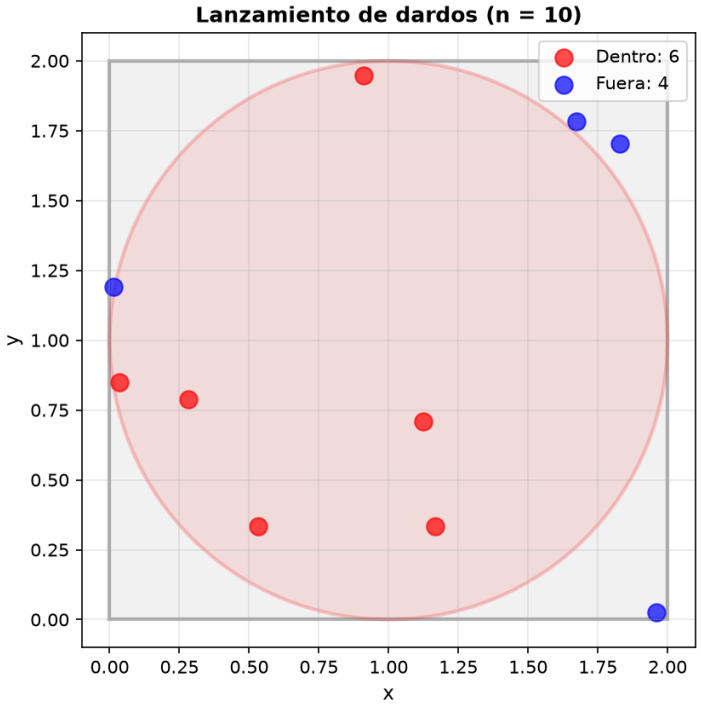
Despejando π:

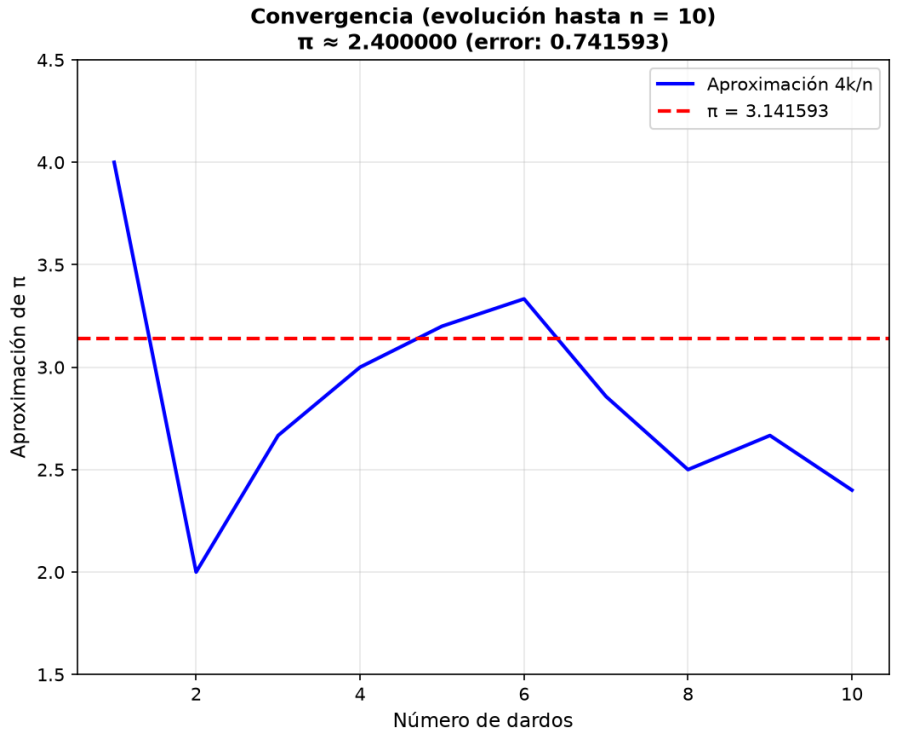
$$\pi \approx 4 \cdot \frac{k}{n}$$

Algoritmo

1. Lanzar n dardos con coordenadas (x_i, y_i) uniformes en $[0, 2] \times [0, 2]$
2. Contar cuántos satisfacen $(x_i - 1)^2 + (y_i - 1)^2 \leq 1$
3. Aproximar $\pi \approx 4k/n$

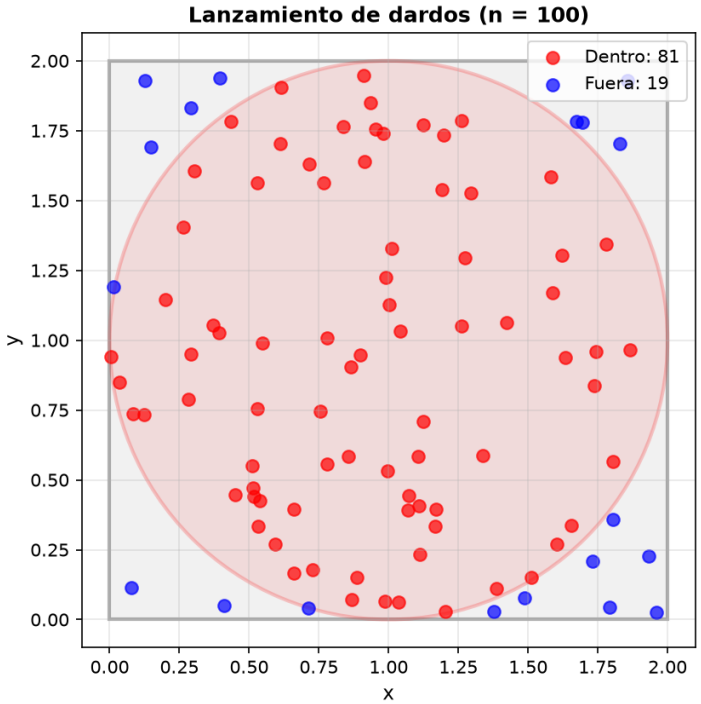
Evolución del Montecarlo: n = 10

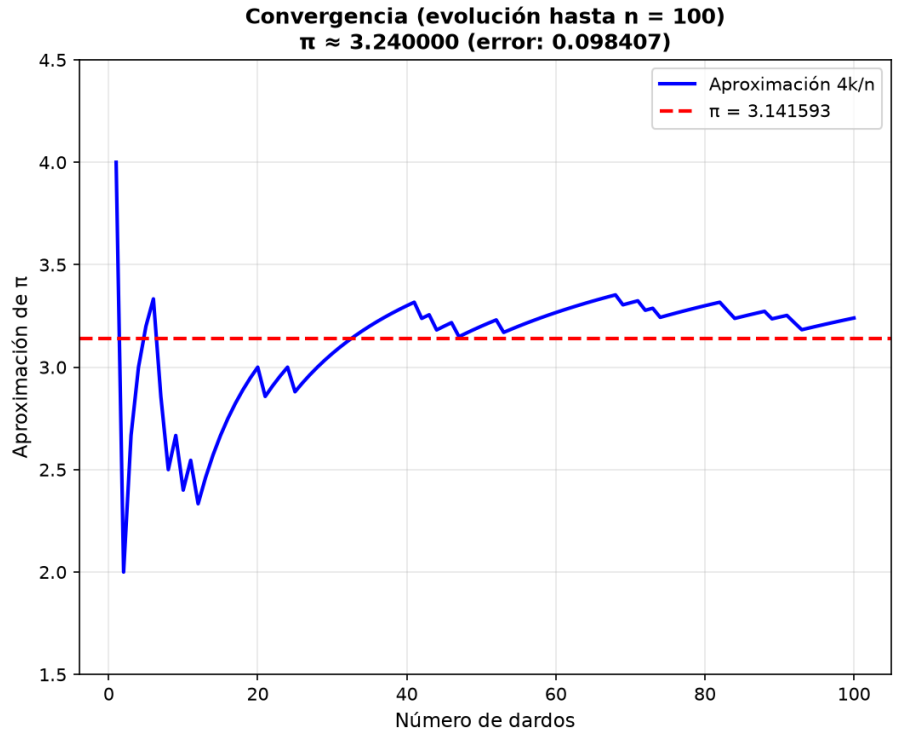




Con apenas 10 dardos, la aproximación es muy inestable. El error es grande.

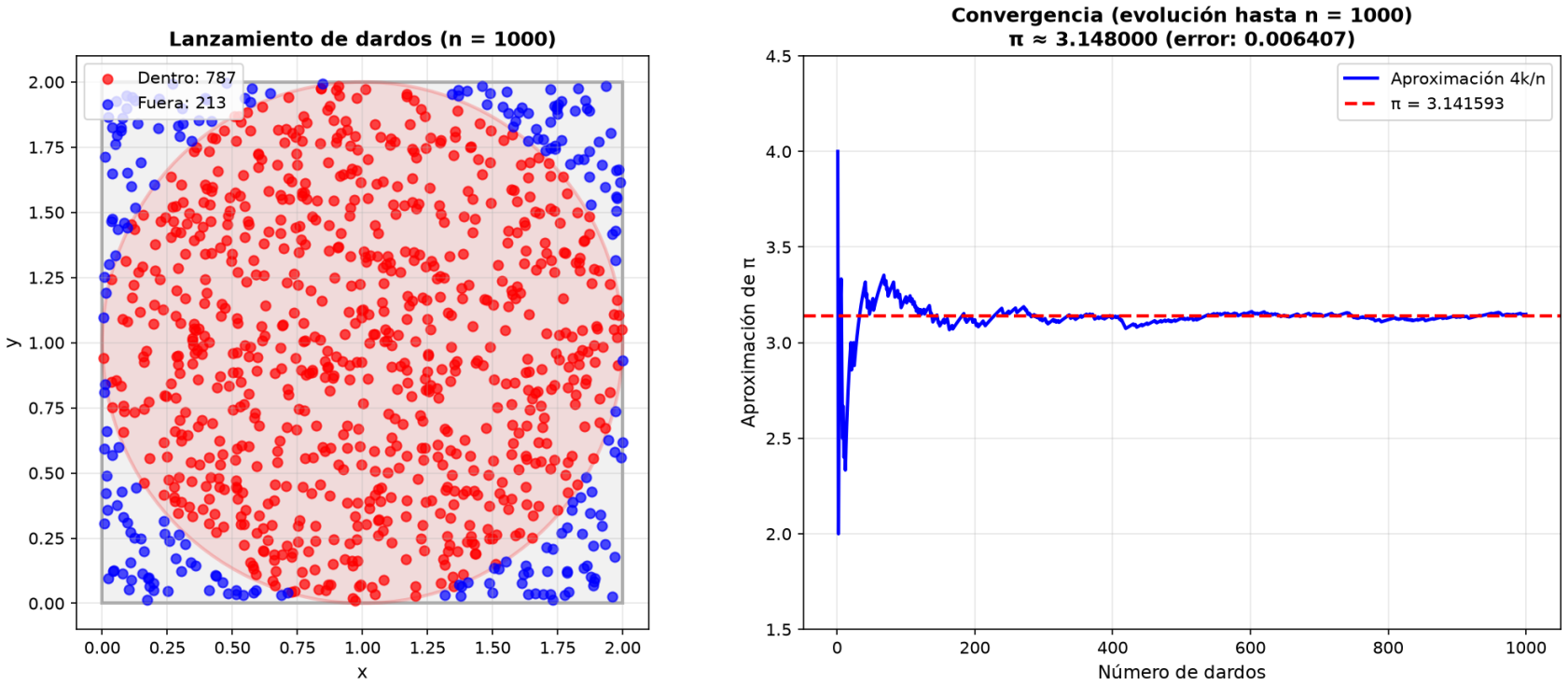
Evolución del Montecarlo: n = 100





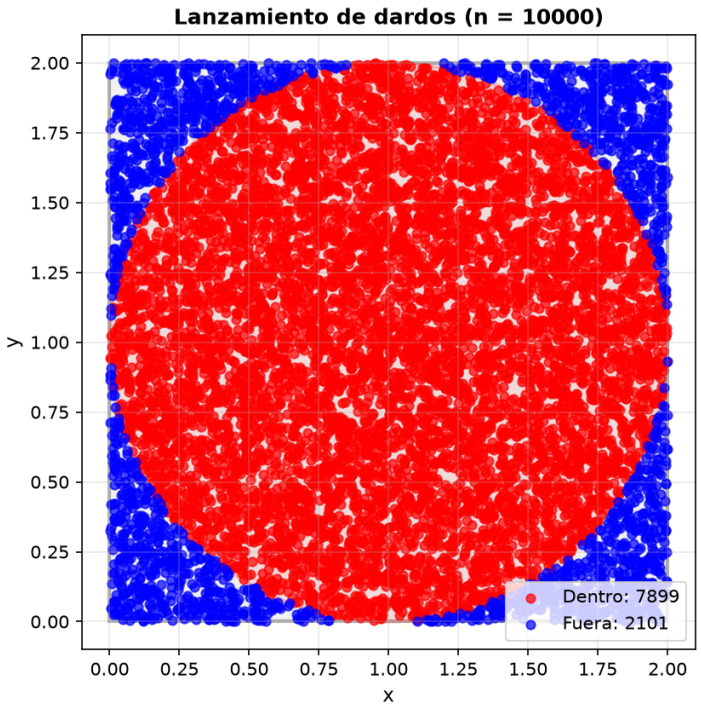
Con 100 dardos, la convergencia es lenta. Se empieza a estabilizar.

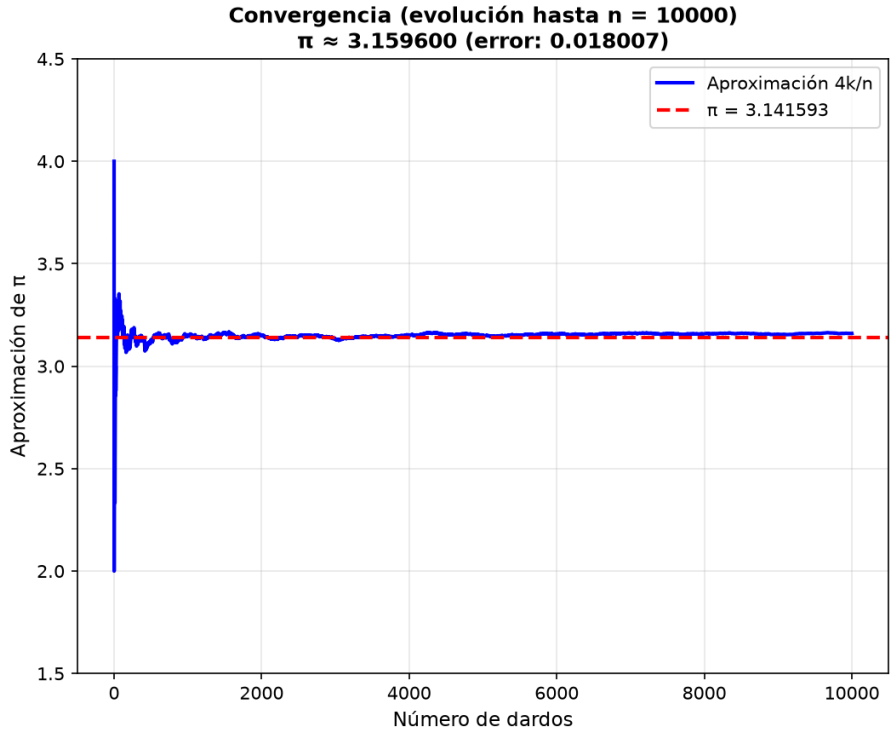
Evolución del Montecarlo: n = 1000



Con 1000 dardos, la aproximación es mucho mejor. El error disminuye significativamente.

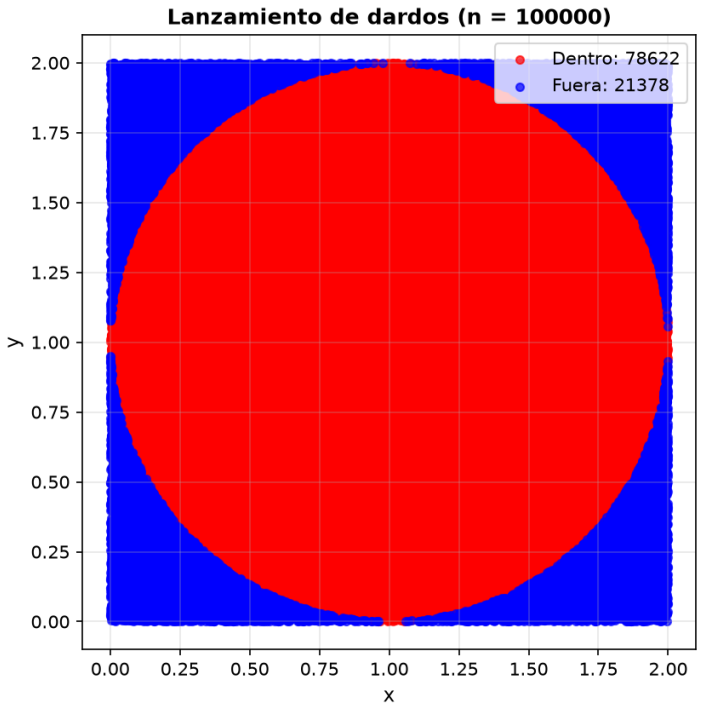
Evolución del Montecarlo: n = 10000

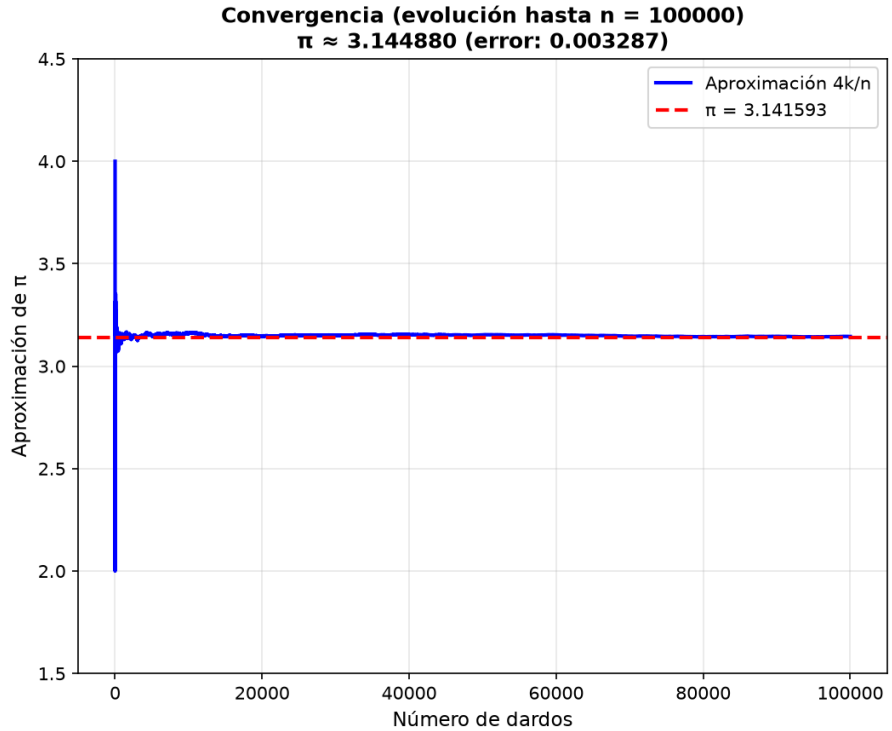




Con 10,000 dardos, el círculo se ve perfectamente delineado. La convergencia es clara.

Evolución del Montecarlo: n = 100000





Con 100,000 dardos, la aproximación es excelente. El círculo es perfectamente visible.

Outline

1. Repaso rápido de probabilidad
2. Ejercicio divertido
3. Montecarlo
4. Las Vegas
5. Estructuras de datos aleatorias
6. Conclusiones y cierre

Algoritmos Montecarlo

Algoritmo Montecarlo

Un algoritmo Montecarlo es un algoritmo aleatorio que puede producir una respuesta incorrecta con cierta probabilidad, pero su tiempo de ejecución es determinístico.

Ejemplo: El método de aproximación de π que vimos es un ejemplo de algoritmo Montecarlo, ya que puede producir una aproximación incorrecta dependiendo del número de dardos lanzados.

Problema tomado en la Meta Hacker Cup

Problema Dado un conjunto de $n \leq 10^5$ hormigas ubicadas en el plano, determinar si existen al menos $\frac{n}{2}$ hormigas que se encuentren alineadas.

Solución al problema de las hormigas

Idea Trivial Para cada hormiga, calcular la pendiente de la línea que conecta esa hormiga con todas las demás. Si alguna pendiente aparece al menos $\frac{n}{2}$ veces, entonces hay al menos $\frac{n}{2}$ hormigas alineadas.

Solución al problema de las hormigas

Idea Trivial Para cada hormiga, calcular la pendiente de la línea que conecta esa hormiga con todas las demás. Si alguna pendiente aparece al menos $\frac{n}{2}$ veces, entonces hay al menos $\frac{n}{2}$ hormigas alineadas.

Complejidad de la idea trivial Para cada hormiga, calcular las pendientes con respecto a todas las demás hormigas toma $O(n)$ tiempo. Como hay n hormigas, el tiempo total es $O(n^2)$. Esto es demasiado lento para n grande.

Solución al problema de las hormigas

Algoritmo Montecarlo

1. Elegir una hormiga al azar.

Solución al problema de las hormigas

Algoritmo Montecarlo

1. Elegir una hormiga al azar.
2. Calcular las pendientes con respecto a todas las demás hormigas.

Solución al problema de las hormigas

Algoritmo Montecarlo

1. Elegir una hormiga al azar.
2. Calcular las pendientes con respecto a todas las demás hormigas.
3. Contar cuántas veces aparece cada pendiente.

Solución al problema de las hormigas

Algoritmo Montecarlo

1. Elegir una hormiga al azar.
2. Calcular las pendientes con respecto a todas las demás hormigas.
3. Contar cuántas veces aparece cada pendiente.
4. Si alguna pendiente aparece al menos $\frac{n}{2}$ veces, devolver "Sí". De lo contrario, repetir el proceso con otra hormiga.

Solución al problema de las hormigas

Algoritmo Montecarlo

1. Elegir una hormiga al azar.
2. Calcular las pendientes con respecto a todas las demás hormigas.
3. Contar cuántas veces aparece cada pendiente.
4. Si alguna pendiente aparece al menos $\frac{n}{2}$ veces, devolver "Sí". De lo contrario, repetir el proceso con otra hormiga.
5. Repetir el proceso un número fijo de veces (por ejemplo, 30) para aumentar la probabilidad de encontrar una solución si existe.

Solución al problema de las hormigas

¿Por qué esto funciona?

Solución al problema de las hormigas

¿Por qué esto funciona?

Si hay al menos $\frac{n}{2}$ hormigas alineadas, entonces la probabilidad de elegir una de esas hormigas es al menos $\frac{1}{2}$. Por lo tanto, si repetimos el proceso varias veces, la probabilidad de no encontrar una solución disminuye exponencialmente.

Solución al problema de las hormigas

¿Por qué esto funciona?

Si hay al menos $\frac{n}{2}$ hormigas alineadas, entonces la probabilidad de elegir una de esas hormigas es al menos $\frac{1}{2}$. Por lo tanto, si repetimos el proceso varias veces, la probabilidad de no encontrar una solución disminuye exponencialmente.

Para 30 hormigas al azar, la probabilidad de no encontrar una solución si existe es:

$$\left(\frac{1}{2}\right)^{30} \approx 9.31 \times 10^{-10}$$

El cuál es una probabilidad muy baja, por lo que podemos estar bastante seguros de que si existe una solución, la encontraremos.

Solución al problema de las hormigas

¿Cuál es una probabilidad aceptable de error?

En realidad, es dificil determinar un número exacto de repeticiones para garantizar una probabilidad de error aceptable. Sin embargo, en la práctica, se puede elegir un número de repeticiones que sea suficientemente grande para que la probabilidad de error sea muy baja y el algoritmo sea eficiente.

Por ejemplo, 30 repeticiones es un número razonable para este problema, dado que el tiempo de ejecución es $T(n) = 30 \cdot n$, lo cual es eficiente para $n \leq 10^5$.

Pero podemos notar que también podemos hacer 100 iteraciones, y no estaríamos poniendo en riesgo la eficiencia del algoritmo, y la probabilidad de error sería aún menor.

Otro truco que se puede hacer

Si tenemos dudas de si la probabilidad de error es aceptable, podemos hacer un truco adicional. Iteramos hasta no superar el Time Limit del problema.

Código en C++

```
// Tiempo máximo de ejecución en segundos
double tmax = 5.0;
// Guardamos el instante inicial
auto start = chrono::high_resolution_clock::now();
double t_eje = 0.0;

while (t_eje < tmax) {
    // --- ACÁ VA EL CÓDIGO /
    auto now = chrono::high_resolution_clock::now();
    t_eje = chrono::duration<double>(now - start).count();
}
```

¿Cómo generar números aleatorios?

Problema En muchos problemas de programación competitiva, necesitamos generar números aleatorios. Sin embargo, la función rand() de C++ no es muy buena y puede ser lenta.

¿Cómo generar números aleatorios?

Problema En muchos problemas de programación competitiva, necesitamos generar números aleatorios. Sin embargo, la función rand() de C++ no es muy buena y puede ser lenta.

Solución Usar un generador de números aleatorios más rápido y de mejor calidad, como el Mersenne Twister (mt19937) que está disponible en C++11.

¿Cómo generar números aleatorios?

Ejemplo de uso

- Inicializar el generador con una semilla:

```
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
```

- Generar un número aleatorio en un rango [a, b]:

```
uniform_int_distribution<int> dist(a, b);
int random_number = dist(rng);
```

¿Cómo generar números aleatorios?

Ejemplo de uso

- Inicializar el generador con una semilla:

```
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
```

- Generar un número aleatorio en un rango [a, b]:

```
uniform_int_distribution<int> dist(a, b);
int random_number = dist(rng);
```

Importante uso de la semilla basada en el tiempo actual para asegurar que los números generados sean diferentes en cada ejecución del programa.

En rondas de Codeforces, usar una semilla fija facilita los hackeos de otros participantes.

Outline

1. Repaso rápido de probabilidad
2. Ejercicio divertido
3. Montecarlo
4. Las Vegas
5. Estructuras de datos aleatorias
6. Conclusiones y cierre

Algoritmos Las Vegas

Algoritmo Las Vegas

Un algoritmo Las Vegas es un algoritmo aleatorio que siempre produce la respuesta correcta, pero su tiempo de ejecución puede variar dependiendo de la aleatoriedad.

Algoritmos Las Vegas

Algoritmo Las Vegas

Un algoritmo Las Vegas es un algoritmo aleatorio que siempre produce la respuesta correcta, pero su tiempo de ejecución puede variar dependiendo de la aleatoriedad.

Ejemplo: El algoritmo de QuickSort aleatorio es un ejemplo de algoritmo Las Vegas, ya que siempre ordena correctamente el arreglo, pero el tiempo de ejecución depende de la elección del pivote.

Problema ejemplo

Problema Dado un programa que tiene un conjunto de $n \leq 1500$ puntos misteriosos en el plano, del cual no sabemos sus coordenadas. En cada operación se le puede consultar por 3 del conjunto. Si existe al menos un punto misterioso que se encuentra dentro del triángulo formado por los 3 puntos consultados, el programa devuelve el índice de alguno de ellos, de lo contrario devuelve "No". Hallar 3 puntos misteriosos de manera tal que la respuesta del programa sea "No" haciendo a lo sumo 75 preguntas.

Problema ejemplo

Problema Dado un programa que tiene un conjunto de $n \leq 1500$ puntos misteriosos en el plano, del cual no sabemos sus coordenadas. En cada operación se le puede consultar por 3 del conjunto. Si existe al menos un punto misterioso que se encuentra dentro del triángulo formado por los 3 puntos consultados, el programa devuelve el índice de alguno de ellos, de lo contrario devuelve "No". Hallar 3 puntos misteriosos de manera tal que la respuesta del programa sea "No" haciendo a lo sumo 75 preguntas.

Este problema se llama Empty Triangle y fue tomado en el primer Contest de TC ARG 2025

Solución al problema

En el primer paso, elegimos 3 puntos aleatorios. Si el programa nos devuelve un índice, significa que ese punto está ubicado dentro de los 3 que hicimos la query.

Notemos que ese punto nos divide al triángulo en otros 3 triángulos. Podemos suponer que el resto de los puntos se van a ubicar distribuidos entre estos 3 triángulos.

Es fácil notar que al menos uno de estos 3 triángulos tendrá una cantidad de puntos menor a $\frac{n}{3}$. Entonces tenemos una probabilidad de $\frac{1}{3}$ de elegir un triángulo con menos de $\frac{n}{3}$ puntos.

Solución al problema

Repetimos el proceso hasta encontrar el "No". Sabemos que vamos a llegar porque en cada momento nos estamos quedando con una cantidad menor de puntos misterios en el interior.

Notemos que el comportamiento lo podemos pensar como una moneda que lanzamos y tiene una probabilidad de $\frac{1}{3}$ de salir cara. Si sale cara, nos quedamos con menos de $\frac{n}{3}$ puntos misteriosos. Si sale cruz, nos quedamos con más de $\frac{n}{3}$ puntos misteriosos.

La esperanza de cantidad de caras luego de 75 queries es $75 \cdot \frac{1}{3} = 25$. Por lo tanto, en promedio, luego de 75 queries vamos a tener una cantidad de puntos misteriosos menor a $\frac{n}{3^{25}}$, lo cual es menor a 1.

Outline

1. Repaso rápido de probabilidad
2. Ejercicio divertido
3. Montecarlo
4. Las Vegas
5. Estructuras de datos aleatorias
6. Conclusiones y cierre

Treap

Un Treap es una estructura de datos que combina un árbol binario de búsqueda (BST) y un heap. Cada nodo tiene una clave y una prioridad aleatoria. La clave sigue la propiedad del BST, mientras que la prioridad sigue la propiedad del heap.

Operaciones:

- Inserción: Se inserta como en un BST y luego se ajusta para mantener la propiedad del heap.
- Separar: Se realiza un corte realizando la búsqueda como en un BST y luego se ajusta para mantener la propiedad del heap.
- Búsqueda: Igual que en un BST.

La aleatoriedad de las prioridades asegura que el árbol esté balanceado en promedio, lo que permite operaciones eficientes.

Implicit Treap

Un Implicit Treap es una variante del Treap que permite manejar arreglos de elementos de manera eficiente. Cada nodo representa un elemento del arreglo y se mantiene el orden de los elementos según su posición en el arreglo.

La estructura permite realizar operaciones típicas de Segment Tree y muchas más, como cortar un pedazo del arreglo y pegarlo en otra posición, o invertir un rango de elementos, todo en tiempo logarítmico en promedio.

Outline

1. Repaso rápido de probabilidad
2. Ejercicio divertido
3. Montecarlo
4. Las Vegas
5. Estructuras de datos aleatorias
6. Conclusiones y cierre

Conclusiones

- No tenerle miedo al Worst Case si se calcula la probabilidad de que ocurra. Muchas veces es muy baja y podemos ignorarla.
- Montecarlo fija el tiempo de ejecución y la probabilidad de error debe ser baja. Las Vegas fija la respuesta correcta y el tiempo de ejecución debe tener un valor esperado acorde a la complejidad del problema.
- Tener bien frescos los conceptos de probabilidad y estadística es fundamental para poder analizar algoritmos aleatorios.
- En ICPC, muchos problemas tienen olfato a ser randomizables. No está mal mandar un manotazo de ahogado si las cuentas de proba no salen.
- Si utilizan la semilla del tiempo actual para generar números aleatorios, el mismo código puede dar veredictos distintos en distintas ejecuciones.

¡Gracias!

¿Consultas?