

Algoritmos aleatorizados

IIC2283

Vamos a permitir a los algoritmos tener una componente aleatoria

- ▶ En general esto significa que un algoritmo toma algunas decisiones dependiendo de valores escogidos al azar (según una distribución de probabilidades)

Algoritmos aleatorizados

Vamos a permitir a los algoritmos tener una componente aleatoria

- ▶ En general esto significa que un algoritmo toma algunas decisiones dependiendo de valores escogidos al azar (según una distribución de probabilidades)

Hablamos entonces de **algoritmos aleatorizados**

Algoritmos aleatorizados

La ejecución de un algoritmo aleatorizado depende entonces de valores escogidos al azar

- ▶ Distintas ejecuciones pueden dar resultados distintos

Algoritmos aleatorizados

La ejecución de un algoritmo aleatorizado depende entonces de valores escogidos al azar

- ▶ Distintas ejecuciones pueden dar resultados distintos

Vamos a considerar dos tipos de algoritmos aleatorizados:

Algoritmos aleatorizados

La ejecución de un algoritmo aleatorizado depende entonces de valores escogidos al azar

- ▶ Distintas ejecuciones pueden dar resultados distintos

Vamos a considerar dos tipos de algoritmos aleatorizados:

- ▶ **Monte Carlo:** el algoritmo siempre entrega un resultado, pero hay una probabilidad de que sea incorrecto

Algoritmos aleatorizados

La ejecución de un algoritmo aleatorizado depende entonces de valores escogidos al azar

- ▶ Distintas ejecuciones pueden dar resultados distintos

Vamos a considerar dos tipos de algoritmos aleatorizados:

- ▶ **Monte Carlo:** el algoritmo siempre entrega un resultado, pero hay una probabilidad de que sea incorrecto
- ▶ **Las Vegas:** si el algoritmo entrega un resultado es correcto, pero hay una probabilidad de que no entregue resultado

¿Cuáles son las ventajas de los algoritmos aleatorizados?

¿Cuáles son las ventajas de los algoritmos aleatorizados?

Existen problemas para los cuales los algoritmos aleatorizados son más eficientes que los algoritmos usuales (sin una componente aleatoria)

- ▶ Por ejemplo, el problema de verificar si un número es primo

¿Cuáles son las ventajas de los algoritmos aleatorizados?

Existen problemas para los cuales los algoritmos aleatorizados son más eficientes que los algoritmos usuales (sin una componente aleatoria)

- ▶ Por ejemplo, el problema de verificar si un número es primo

Existen problemas para los cuales los únicos algoritmos eficientes conocidos son aleatorizados

- ▶ Por ejemplo, el problema de verificar si dos polinomios en varias variables son equivalentes

¿Cuáles son las ventajas de los algoritmos aleatorizados?

Existen problemas para los cuales los algoritmos aleatorizados son más eficientes que los algoritmos usuales (sin una componente aleatoria)

- ▶ Por ejemplo, el problema de verificar si un número es primo

Existen problemas para los cuales los únicos algoritmos eficientes conocidos son aleatorizados

- ▶ Por ejemplo, el problema de verificar si dos polinomios en varias variables son equivalentes

Vamos a ver en detalle estos ejemplos ...

Algoritmos de Monte Carlo: equivalencia de polinomios

Consideramos polinomios en $\mathbb{Q}$

Suponemos inicialmente que un polinomio es una expresión de la forma:

$$p(x) = \sum_{i=1}^k \prod_{j=1}^{\ell_i} (a_{i,j}x + b_{i,j})$$

donde cada $a_{i,j}, b_{i,j} \in \mathbb{Q}$

La forma canónica de $p(x)$ es una expresión de la forma:

$$p(x) = \sum_{i=0}^{\ell} c_i x^i$$

donde cada $c_i \in \mathbb{Q}$ y $\ell \leq \max\{\ell_1, \dots, \ell_k\}$

Si $c_\ell \neq 0$, entonces $p(x)$ no es el polinomio nulo y su grado es ℓ

Algoritmos de Monte Carlo: equivalencia de polinomios

Evaluamos polinomios sobre elementos de $\mathbb{Q}$

- ▶ La operación más cara en este proceso es la multiplicación de números en $\mathbb{Q}$, por eso nos concentramos en ella

Algoritmos de Monte Carlo: equivalencia de polinomios

Evaluamos polinomios sobre elementos de $\mathbb{Q}$

- La operación más cara en este proceso es la multiplicación de números en $\mathbb{Q}$, por eso nos concentramos en ella

Ejercicio

1. De un algoritmo que, dado un polinomio $p(x)$ y $a \in \mathbb{Q}$, calcule $p(a)$
2. De un algoritmo que, dado un polinomio $p(x)$ en su forma canónica y $a \in \mathbb{Q}$, calcule $p(a)$

Los dos algoritmos deben realizar $O(|p(x)|)$ multiplicaciones, donde $|p(x)|$ es el largo de $p(x)$ considerado como una palabra sobre un cierto alfabeto.

Algoritmos de Monte Carlo: equivalencia de polinomios

Dados dos polinomios $p(x)$ y $q(x)$, queremos verificar si son idénticos.

- ▶ Para cada $a \in \mathbb{Q}$, se tiene que $p(a) = q(a)$

Algoritmos de Monte Carlo: equivalencia de polinomios

Dados dos polinomios $p(x)$ y $q(x)$, queremos verificar si son idénticos.

- ▶ Para cada $a \in \mathbb{Q}$, se tiene que $p(a) = q(a)$

¿Cómo podemos resolver este problema?

Un algoritmo para la equivalencia de polinomios

EquivPol($p(x)$, $q(x)$)

transforme $p(x)$ es su forma canónica $\sum_{i=0}^k c_i x^i$

transforme $q(x)$ es su forma canónica $\sum_{i=0}^{\ell} d_i x^i$

if $k \neq \ell$ **then return** no

else

for $i := 0$ **to** k **do**

if $c_i \neq d_i$ **then return** no

return sí

Un algoritmo para la equivalencia de polinomios

Ejercicio

Muestre que el algoritmo anterior en el peor caso es $O(n^2)$, donde $n = |p(x)| + |q(x)|$

- Recuerde que la operación básica a contar es la multiplicación de números racionales

Un algoritmo para la equivalencia de polinomios

Ejercicio

Muestre que el algoritmo anterior en el peor caso es $O(n^2)$, donde $n = |p(x)| + |q(x)|$

- Recuerde que la operación básica a contar es la multiplicación de números racionales

¿Es posible resolver este problema utilizando un menor número de multiplicaciones?

Un algoritmo aleatorizado para la equivalencia de polinomios

Suponga que:

$$p(x) = \sum_{i=1}^k \prod_{j=1}^{r_i} (a_{i,j}x + b_{i,j})$$

$$q(x) = \sum_{i=1}^{\ell} \prod_{j=1}^{s_i} (c_{i,j}x + d_{i,j})$$

Un algoritmo aleatorizado para la equivalencia de polinomios

Suponga que:

$$p(x) = \sum_{i=1}^k \prod_{j=1}^{r_i} (a_{i,j}x + b_{i,j})$$
$$q(x) = \sum_{i=1}^{\ell} \prod_{j=1}^{s_i} (c_{i,j}x + d_{i,j})$$

Utilizamos el siguiente algoritmo **aleatorizado**:

EquivPolAleatorizado($p(x)$, $q(x)$)

$K := 1 + \max\{r_1, \dots, r_k, s_1, \dots, s_{\ell}\}$

escoja al azar y con distribución uniforme un elemento a
del conjunto de números naturales $\{1, \dots, 100 \cdot K\}$

if $p(a) = q(a)$ **then return** sí

else return no

Un algoritmo aleatorizado para la equivalencia de polinomios

El algoritmo sólo necesita realizar $O(n)$ multiplicaciones, donde $n = |p(x)| + |q(x)|$

- ▶ Ya que necesita calcular $p(a)$ y $q(a)$

Un algoritmo aleatorizado para la equivalencia de polinomios

El algoritmo sólo necesita realizar $O(n)$ multiplicaciones, donde $n = |p(x)| + |q(x)|$

- ▶ Ya que necesita calcular $p(a)$ y $q(a)$

Pero el algoritmo puede dar una respuesta equivocada

- ▶ ¿Cuál es la probabilidad de error?

Calculando la probabilidad de error

Sean $p(x)$ y $q(x)$ dos polinomios dados como entrada a

EquivPolAleatorizado

- ▶ Si los polinomios $p(x)$ y $q(x)$ son equivalentes, entonces el algoritmo responde **sí** sin cometer error
- ▶ Si los polinomios $p(x)$ y $q(x)$ no son equivalentes, el algoritmo puede responder **sí** al sacar al azar un elemento $a \in \{1, \dots, 100 \cdot K\}$ tal que $p(a) = q(a)$

Esto significa que a es una raíz del polinomio $r(x) = p(x) - q(x)$

Calculando la probabilidad de error

$r(x)$ no es el polinomio nulo y es de grado a lo más K

- ▶ Por lo tanto $r(x)$ tiene a lo más K raíces en $\mathbb{Q}$

Calculando la probabilidad de error

$r(x)$ no es el polinomio nulo y es de grado a lo más K

► Por lo tanto $r(x)$ tiene a lo más K raíces en $\mathbb{Q}$

Concluimos que:

$$\begin{aligned}\Pr(a \text{ sea una raíz de } r(x)) &\leq \frac{K}{100 \cdot K} \\ &= \frac{1}{100}\end{aligned}$$

Un mejor algoritmo aleatorizado

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ▶ ¿Es aceptable esta probabilidad?

Un mejor algoritmo aleatorizado

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ¿Es aceptable esta probabilidad?

Ejercicio

De un algoritmo que resuelva el problema de equivalencia de polinomios, que en el peor caso sea $O(n)$ y que tenga una probabilidad de error acotada por $\frac{1}{100^{10}}$

Un mejor algoritmo aleatorizado

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ▶ ¿Es aceptable esta probabilidad?

Ejercicio

De un algoritmo que resuelva el problema de equivalencia de polinomios, que en el peor caso sea $O(n)$ y que tenga una probabilidad de error acotada por $\frac{1}{100^{10}}$

¿Confiaría en este algoritmo lineal?

Un mejor algoritmo aleatorizado

La probabilidad de error del algoritmo está acotada por $\frac{1}{100}$

- ¿Es aceptable esta probabilidad?

Ejercicio

De un algoritmo que resuelva el problema de equivalencia de polinomios, que en el peor caso sea $O(n)$ y que tenga una probabilidad de error acotada por $\frac{1}{100^{10}}$

¿Confiaría en este algoritmo lineal?

- ¿Para qué probabilidad estaría dispuesto a confiar?

Una solución para el ejercicio

Suponga que $p(x)$ y $q(x)$ son de la forma definida en la versión anterior de **EquivPolAleatorizado**.

EquivPolAleatorizado($p(x)$, $q(x)$)

$K := 1 + \text{máx}\{r_1, \dots, r_k, s_1, \dots, s_\ell\}$

$A := \{1, \dots, 100 \cdot K\}$

for $i := 1$ **to** 10 **do**

 escoja al azar y con distribución uniforme un elemento a en A

if $p(a) \neq q(a)$ **then return** no

$A := A \setminus \{a\}$

return sí

Un segundo ejemplo: una definición general de polinomios

Consideramos polinomios en varias variables en $\mathbb{Q}$

Un monomio es una expresión de la forma $cx_1^{\ell_1} \cdots x_n^{\ell_n}$, donde $c \in \mathbb{Q}$ y cada $\ell_i \in \mathbb{N}$

Un monomio $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ es nulo si $c = 0$

► No es nulo si $c \neq 0$

El grado de un monomio $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ no nulo es $\ell_1 + \cdots + \ell_n$

Un segundo ejemplo: una definición general de polinomios

Un polinomio es una expresión de la forma:

$$p(x_1, \dots, x_n) = \sum_{i=1}^{\ell} \prod_{j=1}^{m_i} \left(\sum_{k=1}^n a_{i,j,k} x_k + b_{i,j} \right)$$

donde cada $a_{i,j,k} \in \mathbb{Q}$ y cada $b_{i,j} \in \mathbb{Q}$

Un segundo ejemplo: una definición general de polinomios

La forma canónica de un polinomio $p(x_1, \dots, x_n)$ es única, y es igual a 0 o a una suma de monomios que satisface las siguientes propiedades:

- ▶ cada monomio en la forma canónica es de la forma $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ con $c \neq 0$
- ▶ si $cx_1^{\ell_1} \cdots x_n^{\ell_n}$ y $dx_1^{m_1} \cdots x_n^{m_n}$ son dos monomios distintos en la forma canónica, entonces $\ell_i \neq m_i$ para algún $i \in \{1, \dots, n\}$

Un polinomio $p(x_1, \dots, x_n)$ es nulo si su forma canónica es 0

El grado de un polinomio $p(x_1, \dots, x_n)$ no nulo es el mayor grado de los monomios en su forma canónica.

Equivalencia de polinomios en varias variables

Dos polinomios $p(x_1, \dots, x_n)$ y $q(x_1, \dots, x_n)$ son idénticos si para cada secuencia $a_1, \dots, a_n \in \mathbb{Q}$ se tiene que:

$$p(a_1, \dots, a_n) = q(a_1, \dots, a_n)$$

Equivalencia de polinomios en varias variables

Dos polinomios $p(x_1, \dots, x_n)$ y $q(x_1, \dots, x_n)$ son idénticos si para cada secuencia $a_1, \dots, a_n \in \mathbb{Q}$ se tiene que:

$$p(a_1, \dots, a_n) = q(a_1, \dots, a_n)$$

Nuevamente queremos verificar si dos polinomios son idénticos.

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

- Consideramos ahora todas las operaciones, no sólo la multiplicación de números racionales

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

- Consideramos ahora todas las operaciones, no sólo la multiplicación de números racionales

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

- ▶ Consideramos ahora todas las operaciones, no sólo la multiplicación de números racionales

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

- ▶ Incluso si sólo consideramos la multiplicación de números racionales

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

- ▶ Consideramos ahora todas las operaciones, no sólo la multiplicación de números racionales

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

- ▶ Incluso si sólo consideramos la multiplicación de números racionales

Pero existe un algoritmo aleatorizado eficiente para este problema.

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

- ▶ Consideramos ahora todas las operaciones, no sólo la multiplicación de números racionales

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

- ▶ Incluso si sólo consideramos la multiplicación de números racionales

Pero existe un algoritmo aleatorizado eficiente para este problema.

- ▶ Esto no es trivial ya que un polinomio $p(x_1, \dots, x_n)$ puede tener una cantidad infinita de raíces

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

- ▶ Consideramos ahora todas las operaciones, no sólo la multiplicación de números racionales

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

- ▶ Incluso si sólo consideramos la multiplicación de números racionales

Pero existe un algoritmo aleatorizado eficiente para este problema.

- ▶ Esto no es trivial ya que un polinomio $p(x_1, \dots, x_n)$ puede tener una cantidad infinita de raíces
 - ▶ Por ejemplo: $p(x_1, x_2) = (x_1 - 1)(x_2 - 3)$

Equivalencia de polinomios en varias variables

¿Podemos verificar en tiempo polinomial si dos polinomios en varias variables son equivalentes?

- ▶ Consideramos ahora todas las operaciones, no sólo la multiplicación de números racionales

Tenemos un problema: calcular la forma canónica de un polinomio toma tiempo exponencial

- ▶ Incluso si sólo consideramos la multiplicación de números racionales

Pero existe un algoritmo aleatorizado eficiente para este problema.

- ▶ Esto no es trivial ya que un polinomio $p(x_1, \dots, x_n)$ puede tener una cantidad infinita de raíces
 - ▶ Por ejemplo: $p(x_1, x_2) = (x_1 - 1)(x_2 - 3)$
- ▶ El ingrediente esencial es el lema de Schwartz-Zippel

El ingrediente principal

Lema de Schwartz-Zippel

Sea $p(x_1, \dots, x_n)$ un polinomio no nulo de grado k , y sea A un subconjunto finito y no vacío de $\mathbb{Q}$. Si $a_1, \dots, a_n$ son elegidos de manera uniforme e independiente desde A , entonces

$$\Pr(p(a_1, \dots, a_n) = 0) \leq \frac{k}{|A|}$$

Un algoritmo aleatorizado para la equivalencia de polinomios en varias variables

Vamos a dar un algoritmo aleatorizado eficiente para el problema de verificar si dos polinomios en varias variables son equivalentes

Un algoritmo aleatorizado para la equivalencia de polinomios en varias variables

Vamos a dar un algoritmo aleatorizado eficiente para el problema de verificar si dos polinomios en varias variables son equivalentes

Suponga que la entrada del algoritmo está dada por los siguientes polinomios:

$$p(x_1, \dots, x_n) = \sum_{i=1}^{\ell} \prod_{j=1}^{r_i} \left(\sum_{k=1}^n a_{i,j,k} x_k + b_{i,j} \right)$$
$$q(x_1, \dots, x_n) = \sum_{i=1}^m \prod_{j=1}^{s_i} \left(\sum_{k=1}^n c_{i,j,k} x_k + d_{i,j} \right)$$

Un algoritmo aleatorizado para la equivalencia de polinomios en varias variables

EquivPolAleatorizado($p(x_1, \dots, x_n)$, $q(x_1, \dots, x_n)$)

$K := 1 + \text{máx} \{r_1, \dots, r_\ell, s_1, \dots, s_m\}$

$A := \{1, \dots, 100 \cdot K\}$

sea $a_1, \dots, a_n$ una secuencia de números elegidos de
manera uniforme e independiente desde A

if $p(a_1, \dots, a_n) = q(a_1, \dots, a_n)$ **then return** sí

else return no

Utilizando el lema de Schwartz-Zippel

Antes de analizar la complejidad del algoritmo vamos a calcular su probabilidad de error.

Utilizando el lema de Schwartz-Zippel

Antes de analizar la complejidad del algoritmo vamos a calcular su probabilidad de error.

- ▶ Si los polinomios $p(x_1, \dots, x_n)$ y $q(x_1, \dots, x_n)$ son equivalentes, entonces el algoritmo responde **sí** sin cometer error

Utilizando el lema de Schwartz-Zippel

Antes de analizar la complejidad del algoritmo vamos a calcular su probabilidad de error.

- ▶ Si los polinomios $p(x_1, \dots, x_n)$ y $q(x_1, \dots, x_n)$ son equivalentes, entonces el algoritmo responde **sí** sin cometer error
- ▶ Si los polinomios $p(x_1, \dots, x_n)$ y $q(x_1, \dots, x_n)$ no son equivalentes, el algoritmo puede responder **sí** al escoger una secuencia de números $a_1, \dots, a_n$ desde A tales que $p(a_1, \dots, a_n) = q(a_1, \dots, a_n)$
 - ▶ Donde $A = \{1, \dots, 100 \cdot K\}$

Utilizando el lema de Schwartz-Zippel

Antes de analizar la complejidad del algoritmo vamos a calcular su probabilidad de error.

- ▶ Si los polinomios $p(x_1, \dots, x_n)$ y $q(x_1, \dots, x_n)$ son equivalentes, entonces el algoritmo responde **sí** sin cometer error
- ▶ Si los polinomios $p(x_1, \dots, x_n)$ y $q(x_1, \dots, x_n)$ no son equivalentes, el algoritmo puede responder **sí** al escoger una secuencia de números $a_1, \dots, a_n$ desde A tales que $p(a_1, \dots, a_n) = q(a_1, \dots, a_n)$
 - ▶ Donde $A = \{1, \dots, 100 \cdot K\}$

Esto significa que $(a_1, \dots, a_n)$ es una raíz del polinomio $r(x_1, \dots, x_n) = p(x_1, \dots, x_n) - q(x_1, \dots, x_n)$

Utilizando el lema de Schwartz-Zippel

$r(x_1, \dots, x_n)$ no es el polinomio nulo y es de grado t con $t < K$

► Dado que $K = 1 + \max\{r_1, \dots, r_\ell, s_1, \dots, s_m\}$

Utilizando el lema de Schwartz-Zippel

$r(x_1, \dots, x_n)$ no es el polinomio nulo y es de grado t con $t < K$

► Dado que $K = 1 + \max\{r_1, \dots, r_\ell, s_1, \dots, s_m\}$

Utilizando el lema de Schwartz-Zippel obtenemos:

$$\Pr(r(a_1, \dots, a_n) = 0) \leq \frac{t}{|A|} < \frac{K}{|A|} = \frac{K}{100 \cdot K} = \frac{1}{100}$$

Utilizando el lema de Schwartz-Zippel

$r(x_1, \dots, x_n)$ no es el polinomio nulo y es de grado t con $t < K$

► Dado que $K = 1 + \max\{r_1, \dots, r_\ell, s_1, \dots, s_m\}$

Utilizando el lema de Schwartz-Zippel obtenemos:

$$\Pr(r(a_1, \dots, a_n) = 0) \leq \frac{t}{|A|} < \frac{K}{|A|} = \frac{K}{100 \cdot K} = \frac{1}{100}$$

La probabilidad de error del algoritmo está entonces acotada por $\frac{1}{100}$

Un mejor algoritmo aleatorizado para el problema general

Ejercicio

De un algoritmo aleatorizado que resuelva el problema de equivalencia de polinomios en varias variables.

- ▶ La probabilidad de error del algoritmo debe estar acotada por $\frac{1}{100^{10}}$
- ▶ Debe existir una constante c tal que el algoritmo en el peor caso es $O(m^c)$, donde m es el tamaño de la entrada
 - ▶ Si consideramos $p(x_1, \dots, x_n)$ y $q(x_1, \dots, x_n)$ como palabras sobre un cierto alfabeto, entonces $m = |p(x_1, \dots, x_n)| + |q(x_1, \dots, x_n)|$

Una solución para el ejercicio

Suponga que $p(x_1, \dots, x_n)$, $q(x_1, \dots, x_n)$ son de la forma definida en la versión anterior de **EquivPolAleatorizado**.

EquivPolAleatorizado($p(x_1, \dots, x_n)$, $q(x_1, \dots, x_n)$)

$K := 1 + \text{máx} \{r_1, \dots, r_\ell, s_1, \dots, s_m\}$

$A := \{1, \dots, 100 \cdot K + 9 \cdot n\}$

for $i := 1$ **to** 10 **do**

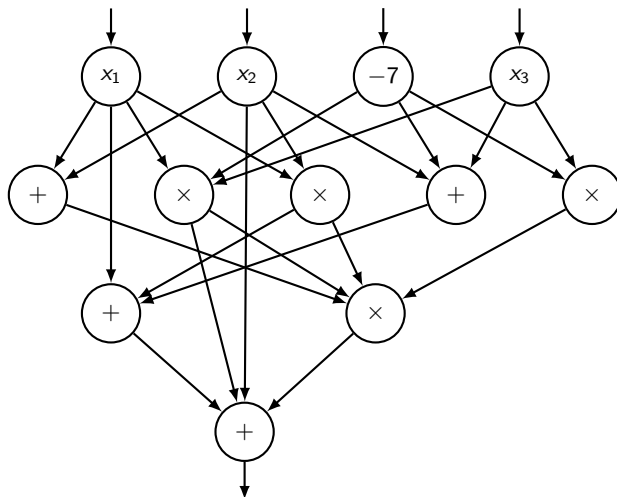
 sea $a_1, \dots, a_n$ una secuencia de números elegidos de
 manera uniforme e independiente desde A

if $p(a_1, \dots, a_n) \neq q(a_1, \dots, a_n)$ **then return no**

$A := A \setminus \{a_1, \dots, a_n\}$

else return sí

Una aplicación: polinomios como circuitos



Algoritmos de Las Vegas: Calculando la mediana de una lista de números enteros

Suponga que n es un número impar y $L[1 \dots n]$ es una lista de números enteros sin elementos repetidos.

- Recuerde que usamos la notación $L[1 \dots n]$ para indicar que L es una lista con n elementos

Algoritmos de Las Vegas: Calculando la mediana de una lista de números enteros

Suponga que n es un número impar y $L[1 \dots n]$ es una lista de números enteros sin elementos repetidos.

- Recuerde que usamos la notación $L[1 \dots n]$ para indicar que L es una lista con n elementos

$L[i]$ es la mediana de L si:

$$|\{j \in \{1, \dots, n\} \mid L[j] < L[i]\}| = \lfloor \frac{n}{2} \rfloor$$

$$|\{k \in \{1, \dots, n\} \mid L[k] > L[i]\}| = \lfloor \frac{n}{2} \rfloor$$

Algoritmos de Las Vegas: Calculando la mediana de una lista de números enteros

Ejercicio

Construya un algoritmo que calcule la mediana de una lista $L[1 \dots n]$ y que en el peor caso sea $O(n \cdot \log_2(n))$

- Considere como la operación básica a contar la comparación de números enteros

Algoritmos de Las Vegas: Calculando la mediana de una lista de números enteros

Ejercicio

Construya un algoritmo que calcule la mediana de una lista $L[1 \dots n]$ y que en el peor caso sea $O(n \cdot \log_2(n))$

- Considere como la operación básica a contar la comparación de números enteros

Vamos a construir un algoritmo aleatorizado de tipo Las Vegas para este problema.

- ¡Este algoritmo funciona en tiempo lineal!

Un algoritmo aleatorizado para el cálculo de la mediana

Suponga que el procedimiento **Mergesort**(L) ordena una lista L utilizando el algoritmo Mergesort

- Recuerde que las listas son pasados por referencia en este curso

Un algoritmo aleatorizado para el cálculo de la mediana

Suponga que el procedimiento **Mergesort**(L) ordena una lista L utilizando el algoritmo Mergesort

- Recuerde que las listas son pasados por referencia en este curso

El siguiente procedimiento calcula la mediana de una lista de enteros $L[1 \dots n]$ (suponiendo que n es impar y L no tiene elementos repetidos):

CalcularMediana($L[1 \dots n]$)

if $n < 2001$ **then**

Mergesort(L)

return $L[\lceil \frac{n}{2} \rceil]$

else

 sea R una lista de $\lceil n^{\frac{3}{4}} \rceil$ números enteros escogido con
 distribución uniforme y de manera independiente desde L

Mergesort(R)

Un algoritmo aleatorizado para el cálculo de la mediana

```
 $d := R[\lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor]$   
 $u := R[\lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil]$   
 $S := \emptyset$   
 $m_d := 0$   
 $m_u := 0$   
for  $i := 1$  to  $n$  do  
    if  $d \leq L[i]$  and  $L[i] \leq u$  then Append( $S, [L[i]]$ )  
    else if  $L[i] < d$  then  $m_d := m_d + 1$   
    else  $m_u := m_u + 1$   
if  $m_d \geq \lceil \frac{n}{2} \rceil$  or  $m_u \geq \lceil \frac{n}{2} \rceil$  or  
     $\text{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor$  then return sin_resultado  
else  
    Mergesort( $S$ )  
    return  $S[\lceil \frac{n}{2} \rceil - m_d]$ 
```

El algoritmo es correcto y eficiente

Ejercicios

Demuestre lo siguiente:

1. Si **CalcularMediana**(L) retorna un número entero m , entonces m es la mediana de L
2. Si se tiene un procedimiento **LanzarMoneda**() que retorna 0 ó 1 con probabilidad $\frac{1}{2}$, entonces existe un algoritmo para construir R que invoca a este procedimiento a los más $c \cdot n^{\frac{3}{4}} \cdot \log_2(n)$ veces, donde c es una constante fija y n es el largo de la lista de entrada
 - Podemos suponer que **LanzarMoneda**() en el peor caso es $O(1)$
3. **CalcularMediana**(L) en el peor caso es $O(n)$, suponiendo que n es el largo de L y considerando todas las operaciones realizadas

¿Cuál es la probabilidad de no retornar un resultado?

La llamada **CalcularMediana**(L) puede no retornar un resultado

- ▶ El procedimiento en este caso retorna *sin_resultado*

Para que **CalcularMediana** pueda ser utilizado en la práctica la probabilidad que no entregue un resultado debe ser baja

¿Cuál es la probabilidad de no retornar un resultado?

La llamada **CalcularMediana**(L) puede no retornar un resultado

- ▶ El procedimiento en este caso retorna *sin_resultado*

Para que **CalcularMediana** pueda ser utilizado en la práctica la probabilidad que no entregue un resultado debe ser baja

- ▶ Vamos a demostrar esto

La probabilidad de no retornar resultado

Sea $L[1 \dots n]$ una lista de números enteros tal que $n \geq 2001$, n es impar y la mediana de L es m

La probabilidad de no retornar resultado

Sea $L[1 \dots n]$ una lista de números enteros tal que $n \geq 2001$, n es impar y la mediana de L es m

Defina las siguientes variables aleatorias:

$$Y_1 = |\{i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\} \mid R[i] \leq m\}|$$

$$Y_2 = |\{i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\} \mid R[i] \geq m\}|$$

La probabilidad de no retornar resultado

Sea $L[1 \dots n]$ una lista de números enteros tal que $n \geq 2001$, n es impar y la mediana de L es m

Defina las siguientes variables aleatorias:

$$Y_1 = |\{i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\} \mid R[i] \leq m\}|$$

$$Y_2 = |\{i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\} \mid R[i] \geq m\}|$$

Estas son variables aleatorias dado que R es construido escogiendo elementos de L con distribución uniforme (y de manera independiente)

La probabilidad de no retornar resultado

Lema

CalcularMediana(L) *retorna sin_resultado si y sólo si alguna de las siguientes condiciones se cumple:*

1. $Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor$
2. $Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil$
3. **Length**(S) $> 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor$

La probabilidad de no retornar resultado

Lema

CalcularMediana(L) *retorna sin_resultado si y sólo si alguna de las siguientes condiciones se cumple:*

1. $Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor$
2. $Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil$
3. **Length**(S) $> 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor$

Ejercicio

Demuestre el lema.

La probabilidad de no retornar resultado

Tenemos entonces que la probabilidad de que **CalcularMediana(L)** retorne *sin_resultado* es igual a:

$$\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor \vee$$

$$Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil \vee \text{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor)$$

La probabilidad de no retornar resultado

Tenemos entonces que la probabilidad de que **CalcularMediana(L)** retorne *sin_resultado* es igual a:

$$\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor) \vee$$

$$Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil \vee \text{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor$$

Necesitamos entonces acotar superiormente $\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor)$,
 $\Pr(Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil)$ y $\Pr(\text{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor)$

La probabilidad de no retornar resultado

Tenemos entonces que la probabilidad de que **CalcularMediana**(L) retorne *sin_resultado* es igual a:

$$\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor) \vee$$

$$Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil \vee \mathbf{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor$$

Necesitamos entonces acotar superiormente $\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor)$,

$\Pr(Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil)$ y $\Pr(\mathbf{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor)$

- Vamos a ver dos desigualdades muy útiles para acotar probabilidades

Teorema

Sea X una variable aleatoria no negativa. Para cada $a \in \mathbb{R}^+$ se tiene que:

$$\Pr(X \geq a) \leq \frac{E(X)}{a}$$

Una demostración de la desigualdad de Markov

Suponemos que el recorrido de X es un conjunto finito $\Omega \subseteq \mathbb{R}_0^+$:

$$\begin{aligned} E(X) &= \sum_{r \in \Omega} r \cdot \Pr(X = r) \\ &= \left(\sum_{r \in \Omega : r < a} r \cdot \Pr(X = r) \right) + \left(\sum_{s \in \Omega : s \geq a} s \cdot \Pr(X = s) \right) \\ &\geq \sum_{s \in \Omega : s \geq a} s \cdot \Pr(X = s) \\ &\geq \sum_{s \in \Omega : s \geq a} a \cdot \Pr(X = s) \\ &= a \cdot \left(\sum_{s \in \Omega : s \geq a} \Pr(X = s) \right) \\ &= a \cdot \Pr(X \geq a) \end{aligned}$$

Una demostración de la desigualdad de Markov

Suponemos que el recorrido de X es un conjunto finito $\Omega \subseteq \mathbb{R}_0^+$:

$$\begin{aligned} E(X) &= \sum_{r \in \Omega} r \cdot \Pr(X = r) \\ &= \left(\sum_{r \in \Omega : r < a} r \cdot \Pr(X = r) \right) + \left(\sum_{s \in \Omega : s \geq a} s \cdot \Pr(X = s) \right) \\ &\geq \sum_{s \in \Omega : s \geq a} s \cdot \Pr(X = s) \\ &\geq \sum_{s \in \Omega : s \geq a} a \cdot \Pr(X = s) \\ &= a \cdot \left(\sum_{s \in \Omega : s \geq a} \Pr(X = s) \right) \\ &= a \cdot \Pr(X \geq a) \end{aligned}$$

Concluimos que $\Pr(X \geq a) \leq \frac{E(X)}{a}$



La desigualdad de Chebyshev

Teorema

$$\Pr(|X - E(X)| \geq a) \leq \frac{\text{Var}(X)}{a^2}$$

Teorema

$$\Pr(|X - E(X)| \geq a) \leq \frac{\text{Var}(X)}{a^2}$$

Demostración. Utilizando la desigualdad de Markov obtenemos:

$$\begin{aligned}\Pr(|X - E(X)| \geq a) &= \Pr((X - E(X))^2 \geq a^2) \\ &\leq \frac{E((X - E(X))^2)}{a^2} \\ &= \frac{\text{Var}(X)}{a^2}\end{aligned}$$



Utilizando la desigualdad de Chebyshev

Lema

$$\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor) \leq n^{-\frac{1}{4}}$$

Utilizando la desigualdad de Chebyshev

Lema

$$\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor) \leq n^{-\frac{1}{4}}$$

Demostración: para cada $i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$, definimos una variable aleatoria X_i de la siguiente forma:

$$X_i = \begin{cases} 1 & R[i] \leq m \\ 0 & R[i] > m \end{cases}$$

Utilizando la desigualdad de Chebyshev

Lema

$$\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor) \leq n^{-\frac{1}{4}}$$

Demostración: para cada $i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$, definimos una variable aleatoria X_i de la siguiente forma:

$$X_i = \begin{cases} 1 & R[i] \leq m \\ 0 & R[i] > m \end{cases}$$

Tenemos que:

$$Y_1 = \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} X_i$$

Utilizando la desigualdad de Chebyshev

Dado que la lista L no contiene elementos repetidos tenemos que:

$$\Pr(X_i = 1) = \frac{\lceil \frac{n}{2} \rceil}{n} = \frac{\frac{n-1}{2} + 1}{n} = \frac{1}{2} + \frac{1}{2 \cdot n}$$

Utilizando la desigualdad de Chebyshev

Dado que la lista L no contiene elementos repetidos tenemos que:

$$\Pr(X_i = 1) = \frac{\lceil \frac{n}{2} \rceil}{n} = \frac{\frac{n-1}{2} + 1}{n} = \frac{1}{2} + \frac{1}{2 \cdot n}$$

De esto se deduce que:

$$\begin{aligned} E(X_i) &= \frac{1}{2} + \frac{1}{2 \cdot n} \\ \text{Var}(X_i) &= \left(\frac{1}{2} + \frac{1}{2 \cdot n} \right) \cdot \left(1 - \frac{1}{2} - \frac{1}{2 \cdot n} \right) \\ &= \left(\frac{1}{2} + \frac{1}{2 \cdot n} \right) \cdot \left(\frac{1}{2} - \frac{1}{2 \cdot n} \right) \\ &= \frac{1}{4} - \frac{1}{4 \cdot n^2} \end{aligned}$$

Utilizando la desigualdad de Chebyshev

Por lo tanto tenemos que:

$$\begin{aligned} E(Y_1) &= E\left(\sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} X_i\right) \\ &= \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} E(X_i) \\ &= \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} \left(\frac{1}{2} + \frac{1}{2 \cdot n}\right) \\ &= \lceil n^{\frac{3}{4}} \rceil \cdot \left(\frac{1}{2} + \frac{1}{2 \cdot n}\right) \end{aligned}$$

Utilizando la desigualdad de Chebyshev

Para $i, j \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$ tal que $i \neq j$ se tiene que X_i es independiente de X_j

Utilizando la desigualdad de Chebyshev

Para $i, j \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$ tal que $i \neq j$ se tiene que X_i es independiente de X_j

Concluimos entonces que:

$$\begin{aligned}\text{Var}(Y_1) &= \text{Var}\left(\sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} X_i\right) \\&= \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} \text{Var}(X_i) \\&= \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} \left(\frac{1}{4} - \frac{1}{4 \cdot n^2}\right) \\&= \lceil n^{\frac{3}{4}} \rceil \cdot \left(\frac{1}{4} - \frac{1}{4 \cdot n^2}\right) \\&\leq \frac{1}{4} \cdot \lceil n^{\frac{3}{4}} \rceil\end{aligned}$$

Utilizando la desigualdad de Chebyshev

Tenemos que:

$$\begin{aligned}\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor) &\leq \Pr(Y_1 < \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}}) \\ &\leq \Pr(Y_1 < \frac{1}{2} \cdot \lceil n^{\frac{3}{4}} \rceil - n^{\frac{1}{2}}) \\ &\leq \Pr(Y_1 < \lceil n^{\frac{3}{4}} \rceil \cdot (\frac{1}{2} + \frac{1}{2 \cdot n}) - n^{\frac{1}{2}}) \\ &= \Pr(Y_1 < E(Y_1) - n^{\frac{1}{2}}) \\ &= \Pr(n^{\frac{1}{2}} < E(Y_1) - Y_1) \\ &\leq \Pr(|Y_1 - E(Y_1)| > n^{\frac{1}{2}}) \\ &\leq \Pr(|Y_1 - E(Y_1)| \geq n^{\frac{1}{2}})\end{aligned}$$

Utilizando la desigualdad de Chebyshev

Por lo tanto, utilizando la desigualdad de Chebyshev concluimos que:

$$\begin{aligned}\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor) &\leq \Pr(|Y_1 - E(Y_1)| \geq n^{\frac{1}{2}}) \\ &\leq \frac{\text{Var}(Y_1)}{n} \\ &\leq \frac{1}{4} \cdot \frac{\lceil n^{\frac{3}{4}} \rceil}{n} \\ &\leq \frac{1}{4} \cdot \frac{n^{\frac{3}{4}} + 1}{n} \\ &\leq \frac{1}{4} \cdot \frac{2 \cdot n^{\frac{3}{4}}}{n} \\ &= \frac{1}{2} \cdot n^{-\frac{1}{4}} \\ &\leq n^{-\frac{1}{4}}\end{aligned}$$



Utilizando nuevamente la desigualdad de Chebyshev

Lema

$$\Pr(Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil) \leq n^{-\frac{1}{4}}$$

Utilizando nuevamente la desigualdad de Chebyshev

Lema

$$\Pr(Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil) \leq n^{-\frac{1}{4}}$$

Ejercicio

Demuestre el lema utilizando las ideas en la demostración del lema anterior.

Un último uso de la desigualdad de Chebyshev

Lema

$$Pr(\mathbf{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor) \leq 4 \cdot n^{-\frac{1}{4}}$$

Un último uso de la desigualdad de Chebyshev

Lema

$$Pr(\mathbf{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor) \leq 4 \cdot n^{-\frac{1}{4}}$$

Demostración. Si $\mathbf{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor$, entonces la menos una de las siguientes condiciones debe ser cierta:

(a) $|\{i \in \{1, \dots, \mathbf{Length}(S)\} \mid S[i] > m\}| \geq 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor$

(b) $|\{i \in \{1, \dots, \mathbf{Length}(S)\} \mid S[i] < m\}| \geq 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor$

Un último uso de la desigualdad de Chebyshev

Vamos a demostrar que la probabilidad de que (a) ocurra es menor o igual a $2 \cdot n^{-\frac{1}{4}}$

Un último uso de la desigualdad de Chebyshev

Vamos a demostrar que la probabilidad de que (a) ocurra es menor o igual a $2 \cdot n^{-\frac{1}{4}}$

De la misma forma se demuestra que la probabilidad que (b) ocurra es menor o igual a $2 \cdot n^{-\frac{1}{4}}$

► De esto se concluye que $\Pr(\mathbf{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor) \leq 4 \cdot n^{-\frac{1}{4}}$

Un último uso de la desigualdad de Chebyshev

Suponga que (a) es cierto, y sea ℓ la posición de u en la lista L ordenada.

- Tenemos que $\ell \geq \lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor$

Un último uso de la desigualdad de Chebyshev

Suponga que (a) es cierto, y sea ℓ la posición de u en la lista L ordenada.

► Tenemos que $\ell \geq \lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor$

Dado que $u = R[\lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil]$, al menos $\lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil$ elementos de R deben estar en posiciones mayores o iguales a ℓ en la lista L ordenada.

Un último uso de la desigualdad de Chebyshev

Suponga que (a) es cierto, y sea ℓ la posición de u en la lista L ordenada.

- Tenemos que $\ell \geq \lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor$

Dado que $u = R[\lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil]$, al menos $\lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil$ elementos de R deben estar en posiciones mayores o iguales a ℓ en la lista L ordenada.

- No podemos asegurar que estos elementos están en posiciones mayores a ℓ puesto que R puede tener elementos repetidos

Un último uso de la desigualdad de Chebyshev

Suponga que (a) es cierto, y sea ℓ la posición de u en la lista L ordenada.

- Tenemos que $\ell \geq \lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor$

Dado que $u = R[\lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil]$, al menos $\lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil$ elementos de R deben estar en posiciones mayores o iguales a ℓ en la lista L ordenada.

- No podemos asegurar que estos elementos están en posiciones mayores a ℓ puesto que R puede tener elementos repetidos
- Vamos a acotar superiormente la probabilidad de que esto ocurra para obtener un cota superior para la probabilidad de que (a) ocurra

Un último uso de la desigualdad de Chebyshev

Para cada $i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$, definimos una variable aleatoria W_i de la siguiente forma:

$$W_i = \begin{cases} 1 & \text{si la posición de } R[i] \text{ en la lista } L \text{ ordenada} \\ & \text{es mayor o igual a } \lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor \\ 0 & \text{en otro caso} \end{cases}$$

Un último uso de la desigualdad de Chebyshev

Para cada $i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$, definimos una variable aleatoria W_i de la siguiente forma:

$$W_i = \begin{cases} 1 & \text{si la posición de } R[i] \text{ en la lista } L \text{ ordenada} \\ & \text{es mayor o igual a } \lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor \\ 0 & \text{en otro caso} \end{cases}$$

Además, definimos la variable aleatoria W como $\sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} W_i$

Un último uso de la desigualdad de Chebyshev

Para cada $i \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$, definimos una variable aleatoria W_i de la siguiente forma:

$$W_i = \begin{cases} 1 & \text{si la posición de } R[i] \text{ en la lista } L \text{ ordenada} \\ & \text{es mayor o igual a } \lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor \\ 0 & \text{en otro caso} \end{cases}$$

Además, definimos la variable aleatoria W como $\sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} W_i$

Dado que $\ell \geq \lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor$, tenemos entonces que la probabilidad de que (a) ocurra es menor o igual a:

$$\Pr(W \geq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil)$$

Un último uso de la desigualdad de Chebyshev

Como L no contiene elementos repetidos obtenemos:

$$\begin{aligned}\Pr(W_i = 1) &= \frac{n - (\lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor) + 1}{n} \\ &= \frac{\lfloor \frac{n}{2} \rfloor - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor + 1}{n} \\ &= \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n}\end{aligned}$$

Un último uso de la desigualdad de Chebyshev

Como L no contiene elementos repetidos obtenemos:

$$\begin{aligned}\Pr(W_i = 1) &= \frac{n - (\lceil \frac{n}{2} \rceil + 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor) + 1}{n} \\ &= \frac{\lfloor \frac{n}{2} \rfloor - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor + 1}{n} \\ &= \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n}\end{aligned}$$

Tenemos entonces que:

$$\begin{aligned}E(W_i) &= \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} \\ \text{Var}(W_i) &= \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} \cdot \left(1 - \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n}\right)\end{aligned}$$

Un último uso de la desigualdad de Chebyshev

Por lo tanto:

$$\begin{aligned} E(W) &= E\left(\sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} W_i\right) \\ &= \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} E(W_i) \\ &= \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} \\ &= \lceil n^{\frac{3}{4}} \rceil \cdot \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} \end{aligned}$$

Un último uso de la desigualdad de Chebyshev

Pero tenemos que:

$$\begin{aligned}\lceil n^{\frac{3}{4}} \rceil \cdot \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} &\leq (n^{\frac{3}{4}} + 1) \cdot \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}} + 3}{n} \\&= n^{\frac{3}{4}} \cdot \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}} + 3}{n} + \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}} + 3}{n} \\&= n^{\frac{3}{4}} \cdot \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}}}{n} + 3 \cdot n^{-\frac{1}{4}} + \frac{1}{2} - 2 \cdot n^{-\frac{1}{4}} + \frac{3}{n} \\&= n^{\frac{3}{4}} \cdot \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}}}{n} + n^{-\frac{1}{4}} + \frac{1}{2} + \frac{3}{n} \\&= \frac{1}{2} \cdot n^{\frac{3}{4}} - 2 \cdot n^{\frac{1}{2}} + n^{-\frac{1}{4}} + \frac{1}{2} + \frac{3}{n} \\&\leq \frac{1}{2} \cdot n^{\frac{3}{4}} - 2 \cdot n^{\frac{1}{2}} + 1\end{aligned}$$

Un último uso de la desigualdad de Chebyshev

Pero tenemos que:

$$\begin{aligned}\lceil n^{\frac{3}{4}} \rceil \cdot \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} &\leq (n^{\frac{3}{4}} + 1) \cdot \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}} + 3}{n} \\&= n^{\frac{3}{4}} \cdot \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}} + 3}{n} + \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}} + 3}{n} \\&= n^{\frac{3}{4}} \cdot \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}}}{n} + 3 \cdot n^{-\frac{1}{4}} + \frac{1}{2} - 2 \cdot n^{-\frac{1}{4}} + \frac{3}{n} \\&= n^{\frac{3}{4}} \cdot \frac{\frac{n}{2} - 2 \cdot n^{\frac{3}{4}}}{n} + n^{-\frac{1}{4}} + \frac{1}{2} + \frac{3}{n} \\&= \frac{1}{2} \cdot n^{\frac{3}{4}} - 2 \cdot n^{\frac{1}{2}} + n^{-\frac{1}{4}} + \frac{1}{2} + \frac{3}{n} \\&\leq \frac{1}{2} \cdot n^{\frac{3}{4}} - 2 \cdot n^{\frac{1}{2}} + 1\end{aligned}$$

Concluimos que $E(W) \leq \frac{1}{2} \cdot n^{\frac{3}{4}} - 2 \cdot n^{\frac{1}{2}} + 1$

Un último uso de la desigualdad de Chebyshev

Para $i, j \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$ tal que $i \neq j$ se tiene que W_i es independiente de W_j

Un último uso de la desigualdad de Chebyshev

Para $i, j \in \{1, \dots, \lceil n^{\frac{3}{4}} \rceil\}$ tal que $i \neq j$ se tiene que W_i es independiente de W_j

Concluimos entonces que:

$$\begin{aligned}\text{Var}(W) &= \text{Var}\left(\sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} W_i\right) \\&= \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} \text{Var}(W_i) \\&= \sum_{i=1}^{\lceil n^{\frac{3}{4}} \rceil} \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} \cdot \left(1 - \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n}\right) \\&= \lceil n^{\frac{3}{4}} \rceil \cdot \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} \cdot \left(1 - \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n}\right) \\&\leq \lceil n^{\frac{3}{4}} \rceil \cdot \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n}\end{aligned}$$

Un último uso de la desigualdad de Chebyshev

Además tenemos que:

$$\begin{aligned}\lceil n^{\frac{3}{4}} \rceil \cdot \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} &\leq \frac{1}{2} \cdot n^{\frac{3}{4}} - 2 \cdot n^{\frac{1}{2}} + 1 \\ &\leq \frac{1}{2} \cdot n^{\frac{3}{4}} + 1 \\ &\leq n^{\frac{3}{4}}\end{aligned}$$

Un último uso de la desigualdad de Chebyshev

Además tenemos que:

$$\begin{aligned}\lceil n^{\frac{3}{4}} \rceil \cdot \frac{\lceil \frac{n}{2} \rceil - 2 \cdot \lfloor n^{\frac{3}{4}} \rfloor}{n} &\leq \frac{1}{2} \cdot n^{\frac{3}{4}} - 2 \cdot n^{\frac{1}{2}} + 1 \\ &\leq \frac{1}{2} \cdot n^{\frac{3}{4}} + 1 \\ &\leq n^{\frac{3}{4}}\end{aligned}$$

Concluimos que $\text{Var}(W) \leq n^{\frac{3}{4}}$

Un último uso de la desigualdad de Chebyshev

Finalmente tenemos que:

$$\begin{aligned}\Pr(W \geq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil) &\leq \Pr(W \geq n^{\frac{3}{4}} - \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} - 1) \\&= \Pr(W \geq \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} - 1) \\&= \Pr(W \geq \frac{1}{2} \cdot n^{\frac{3}{4}} - 2 \cdot n^{\frac{1}{2}} + 1 + n^{\frac{1}{2}} - 2) \\&\leq \Pr(W \geq E(W) + n^{\frac{1}{2}} - 2) \\&\leq \Pr(W \geq E(W) + n^{\frac{1}{2}} - (1 - \frac{1}{\sqrt{2}}) \cdot n^{\frac{1}{2}}) \\&\leq \Pr(W \geq E(W) + \sqrt{\frac{n}{2}}) \\&= \Pr(W - E(W) \geq \sqrt{\frac{n}{2}}) \\&\leq \Pr(|W - E(W)| \geq \sqrt{\frac{n}{2}})\end{aligned}$$

Un último uso de la desigualdad de Chebyshev

Por lo tanto, utilizando la desigualdad de Chebyshev concluimos que:

$$\begin{aligned}\Pr(W \geq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil) &\leq \Pr(|W - E(W)| \geq \sqrt{\frac{n}{2}}) \\ &\leq \frac{\text{Var}(W)}{\frac{n}{2}} \\ &\leq \frac{n^{\frac{3}{4}}}{\frac{n}{2}} \\ &= 2 \cdot n^{-\frac{1}{4}}\end{aligned}$$

Un último uso de la desigualdad de Chebyshev

Por lo tanto, utilizando la desigualdad de Chebyshev concluimos que:

$$\begin{aligned}\Pr(W \geq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil) &\leq \Pr(|W - E(W)| \geq \sqrt{\frac{n}{2}}) \\ &\leq \frac{\text{Var}(W)}{\frac{n}{2}} \\ &\leq \frac{n^{\frac{3}{4}}}{\frac{n}{2}} \\ &= 2 \cdot n^{-\frac{1}{4}}\end{aligned}$$

Concluimos que la probabilidad de que (a) ocurra es menor o igual a $2 \cdot n^{-\frac{1}{4}}$. □

El cálculo final

Recuerde que estamos considerando una lista $L[1 \dots n]$ de números enteros donde n es impar y mayor o igual a 2001.

El cálculo final

Recuerde que estamos considerando una lista $L[1 \dots n]$ de números enteros donde n es impar y mayor o igual a 2001.

Para la lista L demostramos lo siguiente:

$$\begin{aligned}\Pr(Y_1 < \lfloor \frac{1}{2} \cdot n^{\frac{3}{4}} - n^{\frac{1}{2}} \rfloor) &\leq n^{-\frac{1}{4}} \\ \Pr(Y_2 \leq \lceil n^{\frac{3}{4}} \rceil - \lceil \frac{1}{2} \cdot n^{\frac{3}{4}} + n^{\frac{1}{2}} \rceil) &\leq n^{-\frac{1}{4}} \\ \Pr(\mathbf{Length}(S) > 4 \cdot \lfloor n^{\frac{3}{4}} \rfloor) &\leq 4 \cdot n^{-\frac{1}{4}}\end{aligned}$$

El cálculo final

Tenemos entonces que:

$$\Pr(\text{CalcularMediana}(L) \text{ retorne } \textit{sin_resultado}) \leq 6 \cdot n^{-\frac{1}{4}}$$

Tenemos entonces que:

$$\Pr(\text{CalcularMediana}(L) \text{ retorne } \textit{sin_resultado}) \leq 6 \cdot n^{-\frac{1}{4}}$$

Dado que $n \geq 2001$ concluimos que

$$\begin{aligned} \Pr(\text{CalcularMediana}(L) \text{ retorne } \textit{sin_resultado}) &\leq 6 \cdot 2001^{-\frac{1}{4}} \\ &< \frac{9}{10} \end{aligned}$$

El tiempo esperado del algoritmo

Sea p la probabilidad de que **CalcularMediana**(L) retorne resultado

- ▶ Tenemos que $p \geq \frac{1}{10}$

¿En promedio cuántas veces se debe llamar a **CalcularMediana**(L) para obtener la mediana de la lista L ?

El tiempo esperado del algoritmo

Sea T una variable aleatoria tal que para cada $i \geq 1$:

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

Vale decir, T representa el número de llamadas a **CalcularMediana**(L) hasta obtener un resultado

El tiempo esperado del algoritmo

Sea T una variable aleatoria tal que para cada $i \geq 1$:

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

Vale decir, T representa el número de llamadas a **CalcularMediana**(L) hasta obtener un resultado

Dado que T tiene distribución geométrica de parámetro p , concluimos que

$$E(T) = \frac{1}{p} \leq 10$$

El tiempo esperado del algoritmo

Sea T una variable aleatoria tal que para cada $i \geq 1$:

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

Vale decir, T representa el número de llamadas a **CalcularMediana**(L) hasta obtener un resultado

Dado que T tiene distribución geométrica de parámetro p , concluimos que

$$E(T) = \frac{1}{p} \leq 10$$

Concluimos que en promedio se debe llamar 10 veces a **CalcularMediana**(L) para obtener la mediana de la lista L