

IIC2283 — Diseño y Análisis de Algoritmos

Ayudantía 7

Problema 1

En una lista, se define un *elemento mayoritario* como un valor que aparece repetido al menos en la mitad de las posiciones.

Parte 1 Dada una lista de enteros de largo N , desarrolle y analice un algoritmo determinístico que decida si existe un elemento mayoritario, y en ese caso retorne el valor.

Idea de solución Dividir para conquistar. La idea es dividir el arreglo en dos partes, y obtener el elemento mayoritario en cada una. Si hay un elemento mayoritario en el arreglo general, este es mayoritario en alguna de las dos mitades. Luego, basta verificar las dos soluciones posibles, contando el número de ocurrencias de cada una. El algoritmo funciona en tiempo $O(n \log n)$.

Parte 2 Desarrolle y analice un algoritmo aleatorizado que resuelva el mismo problema, pero con un tiempo de ejecución menor.

Solución Generar aleatoriamente un número entre 1 y N , y contar el número de ocurrencias del elemento en esa posición. Si el elemento es mayoritario, se retorna ese elemento, y en caso contrario se retorna que no hay elementos mayoritarios.

Cuando no hay un elemento mayoritario, entonces el algoritmo siempre entrega la respuesta correcta. Por otra parte, si hay un elemento mayoritario, la probabilidad de elegir una posición con otro valor es a lo más $1/2$, por que el elemento mayoritario está repetido al menos en la mitad de las posiciones.

Problema 2

Dadas matrices $A, B, C \in \mathbb{Q}^{n \times n}$, nos interesa determinar si $AB = C$.

Parte 1 Proponga un algoritmo determinístico para resolver el problema, y acote su tiempo de ejecución.

Idea de solución La solución más directa es multiplicar A por B , y comparar todas las posiciones. Esto funciona en tiempo $\omega(n^2)$.

Parte 2 Proponga y analice un algoritmo aleatorizado monte-carlo para el problema, que utilice a lo más n números aleatorios.

Solución Generar un vector x en S^n , donde S es un conjunto de números fijo. Luego se verifica si $Cx = A(Bx)$. Esto se puede realizar en tiempo $\Theta(n^2)$ con el algoritmo usual para multiplicar una matriz por un vector.

Si $AB = C$, entonces para cualquier x se cumple $A(Bx) = Cx$, por lo que en este caso la respuesta es correcta.

Luego, falta acotar la probabilidad de que $A(Bx) = Cx$, cuando $AB \neq C$. Si $AB \neq C$, entonces algunas filas de $(AB - C)$ son distintas de cero, y para todas ellas se cumple $\langle (AB - C)_i, x \rangle = 0$. Si fijamos una fila no nula de $(AB - C)$, denotada por $(AB - C)_i$, entonces $\Pr[(AB - C)x = 0 | AB \neq C] \leq \Pr[\langle (AB - C)_i, x \rangle = 0]$. Esta última probabilidad es igual a la de que un polinomio no nulo de grado uno, con n variables sea igual a 0, al evaluarlo con valores aleatorios. Por el lema de Schartz-Zippel, esto es a lo más $1/|S|$.

Problema 3

Decimos que un árbol dirigido es *no ordenado* si no es importante el orden de los hijos de cada nodo. Dado un árbol no ordenado de altura h , se asocia un polinomio con variables $x_0, x_1, \dots, x_h$ a cada nodo. Estos se definen inductivamente: por cada hoja v , $f_v = x_0$; y por cada nodo interno v , a altura k , con hijos $v_1, \dots, v_m$, se define

$$f_v = (x_k - f_{v_1})(x_k - f_{v_2}) \cdots (x_k - f_{v_m}).$$

Describa y analice un algoritmo aleatorizado que, dados dos árboles dirigidos no ordenados decida si estos son isomorfos. Para esto puede ser útil considerar la definición dada anteriormente, y usar que la factorización de polinomios en polinomios irreducibles es única.

Idea de solución La solución es generar los polinomios según la definición, y luego verificar si los polinomios de las raíces son equivalentes. Para esto se necesita demostrar que estos polinomios serán equivalentes si y solo si los árboles son isomorfos. Ambas direcciones se demuestran por inducción.

Primero se demuestra que si los árboles son isomorfos, entonces los polinomios son equivalentes, por inducción en la altura de los árboles. En el caso base, con árboles de altura cero, el resultado es trivial. Si se cumple hasta altura h , entonces para dos árboles isomorfos de altura $h + 1$, se tiene que cada subárbol de la raíz de uno es isomorfo a un subárbol de la raíz del otro. Por hipótesis de inducción, esto implica que los polinomios de los hijos de la raíz de un árbol son una permutación de los polinomios de los hijos del otro, y por lo tanto por la construcción del polinomio de la raíz, que los polinomios de esta son equivalentes.

En la otra dirección, se sigue por inducción en la variable más alta de los polinomios. Si esta es x_0 , y los polinomios son equivalentes, entonces los únicos árboles posibles son árboles de un solo nodo, que son isomorfos. Por otra parte, si se cumple hasta polinomios con variables $[x_0, \dots, x_h]$, entonces se consideran polinomios con variables $[x_0, \dots, x_h, x_{h+1}]$. Por construcción estos polinomios son equivalentes a polinomios de la forma $(x_{h+1} - p_1) \cdots (x_{h+1} - p_m)$, donde $p_1, \dots, p_m$ solo utilizan las variables $[x_0, \dots, x_h]$. Luego, si hay dos polinomios equivalentes de esta forma: $P = \prod_i (x_{h+1} - p_i)$ y $Q = \prod_i (x_{h+1} - q_i)$, entonces como los polinomios con coeficientes racionales tienen factorización única, se tiene que cada p_i es equivalente a algún q_j . Por hipótesis de inducción, los árboles correspondientes a estos son equivalentes, y por lo tanto los árboles correspondientes a P y Q también lo son.

Finalmente, se utiliza el algoritmo visto en clases para determinar si dos polinomios son equivalentes.