

## Ejercicios Algoritmos Probabilistas y Genéticos

### Algoritmia para Problemas Difíciles (30232)

Departamento de Informática e Ingeniería de Sistemas  
Escuela de Ingeniería y Arquitectura – Universidad de Zaragoza

Noviembre de 2013

#### Ejercicios sobre Algoritmos Probabilistas y Genéticos

---

**Ejercicio 1** Diseñar un algoritmo probabilista numérico para el cálculo aproximado de  $\pi$  basado en el lanzamiento aleatorio de dardos contra una diana circular inscrita en un cuadrado. Experimentar con ese algoritmo.

**Ejercicio 2** Sea  $T(1..n)$  un vector de  $n$  elementos. Se dice que  $x$  es el dato mayoritario en  $T$  si  $|\{i|T(i) = x\}| > n/2$ . Se dice que el vector  $T$  es mayoritario si contiene un elemento mayoritario.

Diseñar un algoritmo de Monte Carlo para determinar si un vector es mayoritario basado en la selección aleatoria de un elemento y la comprobación de si ese elemento es mayoritario o no. Argumentar sobre la probabilidad de error del algoritmo. Basado en el algoritmo anterior, diseñar otro con tiempo de ejecución  $O(n \log(1/e))$ , donde  $n$  es el número de elementos del vector y  $e$  es la probabilidad de error.

Nota: Este ejercicio es sólo interesante como ilustración de los algoritmos de Monte Carlo pues se conoce un algoritmo determinista lineal (piensa en ello).

**Ejercicio 3** Implementar el algoritmo de Las Vegas para el problema de las  $n$  reinas que coloca las  $k$  primeras reinas aleatoriamente antes de intentar completar la solución mediante búsqueda con retroceso. Experimentar con el problema de las 39 reinas para distintos valores de  $k$ . Encontrar una solución para el problema de las 100 reinas y otra para el de las 1000 reinas.

**Ejercicio 4** Una lista compacta de longitud  $n$  se implementa con dos vectores,  $val(1..n)$  y  $ptr(1..n)$ , y un entero, cabeza. El primer elemento de la lista está en  $val(cabeza)$ , el siguiente en  $val(ptr(cabeza))$ , etcétera. En general, si  $val(i)$  no es el último elemento de la lista,  $ptr(i)$  guarda el índice en  $val$  del siguiente elemento de la lista. Si  $val(i)$  es el último elemento de la lista, entonces  $ptr(i) = 0$ .

Considerar una lista compacta cuyos elementos están en orden creciente, es decir,  $val(i) < val(ptr(i))$  para todo  $i = 1, 2, \dots, n$  tal que  $ptr(i) \neq 0$ . Sea  $x$  un elemento. El problema es localizar  $x$  en la lista, es decir, dar su posición en  $val$ . El método de búsqueda dicotómica no es aplicable porque no hay forma directa de localizar el punto intermedio de la lista. Diseñar un algoritmo probabilista más rápido que la búsqueda secuencial para resolver el problema.

**Ejercicio 5** Evaluar y mejorar las páginas de la Wikipedia hispana (<http://es.wikipedia.org/>) sobre el tema. Se trataría de comprobar y mejorar si es preciso lo que aparece en la página, contrastándolo con la bibliografía y con las páginas equivalentes de la Wikipedia inglesa (<http://en.wikipedia.org/>). Naturalmente, esta última también es posible que pueda mejorarse.

**Ejercicio 6** Hemos visto algunos algoritmos probabilistas para realizar pruebas de primalidad para números enteros. Se propone verificar cuál de ellos se utiliza para la generación de claves en algún proyecto de software libre como OpenSSL o GnuPGP.

**Ejercicio 7** Encontrar todos los números entre 0 y 255 que tienen una representación binaria tal que el número total de ceros es igual al de unos. Por ejemplo, 240 es uno de esos números (su representación es

11110000) y 153 es otro (representación: 10011001).

Discutir un algoritmo para encontrar esos números de manera directa y exhaustiva.

Discutir un algoritmo genético para encontrar números que cumplan la condición. ¿Los encuentra todos? Si es así, ¿cómo? Si no, ¿qué habría que hacer para encontrar más números que cumplan la condición?

**Ejercicio 8** Discutir la posibilidad de resolver sudokus mediante algoritmos genéticos.

---

En caso de entregar alguno de estos ejercicios, deberá hacerse hasta el día 20 de diciembre.
--