

Resumen de programación 3

Tema 1. Preliminares.

Índice:

1.1. Introducción	3
1.2. ¿Qué es un algoritmo?	3
1.3. Notación para los programas	5
1.4. Notación matemática	5
1.5. Técnica de demostración 1: Contradicción	5
1.6. Técnica de demostración 2: Inducción matemática	6
1.6.1. El principio de inducción matemática	6
1.7. Recordatorio	7

Bibliografía:

Se ha tomado apuntes del libro:

- *Fundamentos de algoritmia*. G. Brassard y P. Bratley

Previo a ver este resumen decir que conviene leerse este tema, ya que así nos iremos familiarizando con la algoritmia y determinados conceptos, como la demostración por inducción. Este resumen es complementario al libro, por lo que conviene estudiarlo junto.

1.1. Introducción

En los problemas veremos estos pasos a seguir para resolverlos, que será bastante importante el tenerlo claro:

- Elección del esquema (voraz, vuelta atrás, divide y vencerás).
- Identificación del problema con el esquema.
- Estructura de datos.
- Algoritmo completo (con pseudocódigo).
- Estudio del coste.

A lo largo de nuestro temario seguiremos este planteamiento:

- Empezaremos a ver el estudio del coste (temas 2, 3 y 4).
- Luego veremos la estructura de datos (tema 5).
- Para a continuación ver los distintos esquemas, que los dividiremos en 3, siguiendo este mismo orden (es recomendado):
 1. Esquema voraz (tema 6).
 2. Exploración de grafo, que incluyen exploración en anchura, profundidad, vuelta atrás y ramificación y poda (tema 9).
 3. Esquema de divide y vencerás (tema 7).

A continuación, daremos unas nociones básicas previas antes de entrar en nuestra planificación del temario, de nuevo lo hacemos para que vayan sonando los conceptos.

En este tema, así como en el tema 2 no tendremos problemas del mismo. Por lo que se tendrá que estudiar la teoría dada, que aunque es introductoria siempre conviene hacerlo.

1.2. ¿Qué es un algoritmo?

Un **algoritmo** es un conjunto de reglas para efectuar algún cálculo, bien sea a mano o, más frecuentemente, en una máquina. Nos interesan los algoritmos utilizados en una computadora. La ejecución de un algoritmo no debe de implicar ninguna decisión subjetiva ni tampoco debe de hacer preciso el uso de la intuición y de la creatividad.

Es importante decidir cuál de los algoritmos escoger de entre una gama de ellos para resolver un problema. Dependiendo de nuestras prioridades y de los límites del equipo que esté disponible para nosotros, quizá necesitemos seleccionar el algoritmo que requiera menos tiempo, o el que utilice menos espacio, o el que sea más fácil de programar, y así sucesivamente.

La Algoritmia es la ciencia que nos permite evaluar el efecto de estos diferentes factores externos sobre los algoritmos disponibles, de tal modo que sea posible seleccionar el que más se ajuste a nuestras circunstancias particulares; también es la ciencia que nos indica la forma de diseñar un nuevo algoritmo para una tarea concreta.

Ejemplo: multiplicación de dos números enteros. Para hacerlo tendremos estos métodos:

- **Algoritmos “clásicos”:** Son similares a la multiplicación con lápiz y papel de los dos números enteros. Para ello, tendremos la multiplicación inglesa y americana, donde la única diferencia reside en que se multiplica de izquierda a derecha o viceversa.
- **Algoritmo de multiplicación “a la russe”:** Para ello seguiremos este procedimiento:
 1. Se hacen dos columnas poniendo el número menor a la izquierda y el mayor a la derecha. El de la izquierda se divide por 2 hasta llegar a 1 y el de la derecha se multiplica por dos.
 2. Se eliminan las filas en las cuales el número de la columna izquierda sea par y se suman los números que quedan en la columna de la derecha. Es la más parecida a la que se emplea en el hardware de una computadora binaria.
- **Algoritmo mediante divide y vencerás:** Seguiremos estos pasos por este orden (separado respecto al libro para que sea mayor su comprensión):
 1. Necesitamos tener el multiplicando y multiplicador con el mismo número de cifras múltiplo de dos, si no se añadirían ceros, si hiciera falta. Multiplicamos la mitad izquierda del multiplicando por la mitad izquierda del multiplicador y ponemos el resultado desplazado hacia la izquierda tantas veces como cifras haya en el multiplicador: cuatro en nuestro caso.
 2. Luego la mitad izquierda del multiplicando con la derecha del multiplicador y desplazamos dos a la izquierda.
 3. Después, multiplicamos la mitad derecha del multiplicando por la izquierda del multiplicador y desplazamos dos a la izquierda.
 4. Por último, ambas mitades derechas y no desplazamos ninguna posición. Para finalizar, sumamos estos cuatro resultados intermedios.

Hemos reducido la multiplicación de dos números de cuatro cifras a cuatro multiplicaciones de números de cuatro cifras, junto con un cierto número de desplazamientos y una suma final.

De entre los cuatro algoritmos ya vistos, este último reduce la multiplicación de dos números grandes a tres y no cuatro, por lo que sería el más eficiente. La meta de nuestro libro es enseñar a tomar este tipo de decisiones.

1.3. Notación para los programas.

Es importante decidir la forma en que vamos a describir nuestros algoritmos. Para ello no nos limitaremos en ningún lenguaje de programación concreto (lo que denominaremos pseudocódigo): de esta manera, los aspectos esenciales de un algoritmo no resultarán oscurecidos por detalles de programación relativamente poco importantes y no importa cuál sea el lenguaje bien estructurado que prefiere el lector.

Ejemplo: Tenemos esta función con la notación descrita anteriormente:

```
funcion rusa (m, n)
  resultado  $\leftarrow$  0;
  repetir
    si m es impar entonces resultado  $\leftarrow$  resultado + n;
     $m \leftarrow m \div 2$ ;
     $n \leftarrow n + 1$ ;
  hasta que  $m = 1$ ;
  devolver resultado;
```

1.4. Notación matemática.

Se usará la notación matemática para:

- Cálculo proposicional.
- Teoría de conjuntos.
- Enteros, reales e intervalos.
- Funciones y relaciones.
- Cuantificadores.
- Sumas y productos.

1.5. Técnica de demostración 1: Contradicción.

La **demostración por contradicción**, o prueba indirecta, consiste en demostrar la veracidad de una sentencia demostrando que su negación da lugar a una contradicción.

Ejemplo: Se nos da un teorema que dice que existen infinitos números reales, el cual demostraremos por contradicción para ello, realizaremos estos pasos (usaremos esta demostración en los esquemas voraces, por ello es importante el concepto):

- Empezaremos *suponiendo lo contrario* al teorema, en este caso, que existe un conjunto finito, para así buscar una contradicción.
- Si terminamos en una afirmación evidentemente falsa podemos deducir que la primera sentencia es verdadera.

1.6. Técnica de demostración 2: Inducción matemática.

Es una de las herramientas básicas útiles en la Algoritmia. Tendremos dos enfoques básicos opuestos fundamentales:

1. **Inducción:** Consiste en inferir una *ley general a partir de casos particulares*. En general no se puede confiar en el resultado del razonamiento inductivo. Mientras que haya casos que no hayan sido considerados, sigue siendo posible que la regla general inducida sea correcta. Aunque no se puede despreciar este enfoque.

Ejemplo: Consideremos el polinomio $p(n) = n^2 + n + 41$. Si computamos valores como $p(0)$, $p(1)$, $p(2)$, ..., $p(10)$, se va encontrando 41, 43, 47, ... y 151. Es fácil verificar que todos estos enteros son números primos. Por tanto, es natural inferir por inducción que $p(n)$ es primo para todos los valores enteros de n , pero $p(40) = 1.681 = 41^2$ es compuesto. Deducimos entonces que es una *falsa inducción*.

Hay veces que es necesario usar la inducción porque es el único método, como, por ejemplo, en el descubrimiento del cometa Halley, hecho a partir de datos obtenidos en experimentos.

- **Deducción:** Es una inferencia *de lo general a lo particular*. Por contraste, el razonamiento deductivo no está sometido a este tipo de errores. Siempre y cuando la regla invocada sea correcta, y sea aplicable a la situación que se estudia, la conclusión que se alcanza es necesariamente correcta.

Una de las técnicas deductivas más útiles que están disponibles en matemáticas se llama **inducción matemática**. No hay que confundir con el otro enfoque, que aunque se parezca en el nombre no es similar.

1.6.1. El principio de inducción matemática.

Consideremos el siguiente algoritmo:

```
funcion cuadrado (n)
  si  $n = 0$  entonces devolver 0
  si no devolver  $2 * n + \text{cuadrado}(n - 1) - 1$ 
```

Si se hace pruebas en varias entradas pequeñas, observamos que $\text{cuadrado}(0) = 0$; $\text{cuadrado}(1) = 1$; $\text{cuadrado}(2) = 4$;...

Por inducción, parece evidente que $\text{cuadrado}(n) = n^2$ para todos los $n \geq 0$.

Definición: Este principio dice que la propiedad P es cierto para todo $n \geq a$ si:

1. $P(a)$ es cierto.
2. $P(n)$ debe ser cierto siempre que $P(n - 1)$ sea válido para todos los enteros $n > a$.

A veces, las demostraciones mediante inducción matemática se pueden *transformar* en algoritmos, como puede ser el ejemplo del embaldosado.

1.7. Recordatorio.

Se nos dan recordatorios, tales como límites, sumas de series sencillas, combinatoria y probabilidad. En ejercicios posteriores los usaremos, por lo que evitamos dar más detalles.