Algorítmica

2º Grado en Ingeniería Informática

Guión de prácticas

Algoritmos Divide y Vencerás

Algoritmos Divide y Vencerás

1. Objetivo

El objetivo de la práctica consiste en que el alumno sea capaz de analizar un problema, comprobar si puede ser resoluble mediante la técnica de diseño de algoritmos Divide y Vencerás, y aplicarla. Para ello, se expone un conjunto de 5 problemas que deberán ser resueltos por cada grupo de estudiantes.

2. Problemas

2.1. La mayoría absoluta

Dado un vector de enteros V de tamaño n que en cada posicion contiene el código numérico del candidato votado por una persona (hay n votos), se desea conocer si hay algún candidato "x" que tenga mayoría absoluta (puede haber uno solo, o ninguno), es decir necesita más (estrictamente mayor) de n/2 votos. O lo que es lo mismo: $Card\{i \mid v[i]=x\} > n/2$. No se conoce a priori quienes son los candidatos. Por ejemplo si n = 10 tiene que tener 6 votos o mas. Si n = 11, necesita tambien 6 votos o mas. No se puede suponer que exista una relacion de orden entre los elementos del vector.

2.2. Tuercas y tornillos

En una habitacion oscura se tienen dos cajones, en uno de los cuales hay n tornillos de varios tamaños, todos distintos, y en el otro las correspondientes n tuercas. Es necesario emparejar cada tornillo con su tuerca correspondiente, pero debido a la oscuridad no se pueden comparar tornillos con tornillos ni tuercas con tuercas, y la unica comparacion posible es la de intentar enroscar una tuerca en un tornillo para comprobar si es demasiado grande, demasiado pequeña, o se ajusta perfectamente al tornillo.

Para ello, Tenemos dos vectores a y b de tamaño n que contienen los mismos numeros enteros (que son distintos entre sí (no se repiten numeros) pero en diferente orden. Se pretende emparejar los elementos de los dos vectores, de forma que el orden de los elementos en ambos vectores sea el mismo (no importa que orden es, lo importante esque los elementos iguales en ambos vectores esten en la misma posicion, o sea que los dos vectores sean iguales).

La restriccion es que no es posible comparar entre sí los elementos de un mismo vector, solo se pueden realizar comparaciones (de mayor, menor o igual) entre elementos de distintos vectores (por ejemplo se puede preguntar si a[i] < b[j], pero no se puede preguntar si a[i] < a[j]).

2.3. Producto de tres elementos

Determinar si un cierto numero natural N puede expresarse como producto de tres numeros naturales consecutivos.

2.4. Eliminar elementos repetidos

Dado un vector de n elementos, se pide eliminar todos los elementos duplicados, es decir, que estén más de una vez en el vector.

2.5. Organización del calendario e un campeonato

Se organiza un torneo con **n** participantes. Cada participante tiene que competir **exactamente** una vez con todos los posibles oponentes. Además, cada participante tiene que jugar exactamente un partido cada día. Por concreción, y sin perdida de generalidad, puede suponerse que las competiciones se celebran en días sucesivos y que cada participante compite una vez por día. Podemos suponer que el número de participantes es potencia de dos, lo que nos simplificará el problema (no es necesario que haya jornadas de descanso). Por lo tanto n = 2^k participantes, con k entero positivo. Se pide construir un calendario que permita que el torneo concluya en **n-1** días.

3. Tareas a realizar

Para cada problema se pide:

1. Diseñar un método básico (no Divide y Vencerás) que resuelva el problema. Estudiar su eficiencia teórica.
2. Estudiar como el problema puede ser abordado mediante la técnica Divide y Vencerás y realizar el diseño completo. Estudiar su eficiencia teórica.
3. Implementar los algoritmos básico y Divide y Vencerás. Resolver el problema del umbral de forma experimental.

4. Entrega de la práctica

Se deberá entregar un **fichero ZIP** conteniendo:

- Uno o varios ficheros .cpp por cada problema con el código fuente desarrollado y un fichero makefile para compilarlos.
- Una memoria de prácticas en **PDF**, describiendo las soluciones a cada uno de los apartados. La memoria deberá incluir un apartado en el que se indique cómo compilar los algoritmos y cómo ejecutarlos para algún ejemplo de prueba.

La práctica deberá ser entregada por PRADO, en la fecha y hora límite explicada en clase por el profesor. No se aceptarán, bajo ningún concepto, prácticas entregadas con posterioridad a la fecha límite indicada. La entrega de PRADO permanecerá abierta con, al menos, una semana de antelación antes de la fecha límite, por lo que todo alumno tendrá tiempo suficiente para entregarla.

El profesor, en clase de prácticas, podrá realizar auditorías de las prácticas a discreción, con el fin de asegurar de que los estudiantes alcanzan las competencias deseadas. Por este motivo, una vez finalizada la entrega de prácticas por PRADO, es recomendable repasar los ejercicios entregados para poder responder a las preguntas del profesor, llegado el caso de su defensa.