

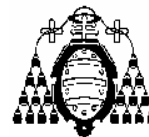


Examen de Teoría de la Programación

E. U. ING. TEC. EN INFORMÁTICA DE OVIEDO

Final Junio – Curso 2004-2005

4 de junio de 2005



DNI _____ Nombre _____ Apellidos _____
Titulación: ☐ Gestión ☐ Sistemas

3. (1 punto) Dado un vector $a[1..n]$, con todos los elementos diferentes y, se supone que, ordenado crecientemente hasta la posición $n-1$, especificar un algoritmo que inserte el elemento $a[n]$ (que también es diferente a los demás elementos) en el lugar que le corresponda del vector $a[1..n]$ de forma que el vector resultante esté ordenado crecientemente.

Ejemplo: (2, 6, 8, 9, 16, 5) \rightarrow (2, 5, 6, 8, 9, 16)

La interfaz de la acción sería la siguiente:

Acción inserta_ordenado(a: ent/sal vector; n: entero)

Donde Tipo vector= vector [1..1000] de entero.

- Escribir la precondition formal que verifique las condiciones de entrada incluyendo los límites de n.
- Escribir la postcondición formal que asegure el correcto funcionamiento de la acción según el enunciado dado.

4. (1,5 puntos) Al final de los años 60, Strassen logró demostrar que las matrices de 2×2 se pueden multiplicar empleando 7 multiplicaciones escalares, en vez de las 8 que se emplean en el algoritmo convencional de la multiplicación de matrices.

Supongamos que tenemos matrices de orden 2^x .

- Cómo aplicarías el algoritmo divide y vencerás para multiplicar estas matrices suponiendo que contamos con las relaciones de Strassen. Explica, simplemente como divides el problema en subproblemas, que llamadas recursivas tenemos y como realizarías la composición de los resultados de cada subproblema.
- Qué complejidad temporal tendría el algoritmo planteado.
- Si tenemos una matriz de 4×4 , ¿Qué algoritmo emplearías, el clásico o el de Strassen? (Razona la respuesta)

5. (2 puntos) Estamos desarrollando un módulo software para las cajas de un supermercado en el que tenemos que dar información detallada sobre la cantidad de monedas (sistema monetario del euro) de cada tipo hay que devolver cuando un cliente paga su compra.

Queremos que el número de monedas que manejan las cajeras sea siempre el mínimo posible, como el número total de monedas nunca será muy alto no nos importa demasiado el tiempo de cálculo. Hay que tener en cuenta que cada mañana, las cajeras meten un número fijo de 10 monedas para cada tipo y que en un momento determinado puede quedarse sin algún tipo de monedas.

- Qué técnica de las vistas en clase emplearías (razona la respuesta en una frase).
- Escribir el código en Java del algoritmo que a partir de las monedas disponibles en ese momento y la cantidad que hay que devolver, permita obtener siempre el número óptimo de monedas o un error en caso de que no sea posible devolver la cantidad. Escribir el código de llamada al método.

6. (2 puntos) Ante la avalancha de gente en la inauguración de la nueva tienda de IKEA en Asturias, los responsables de la tienda están pensando en una forma para minimizar el tiempo total que tardan todos los clientes de un día en realizar su compra.

Para simplificar, han decidido resolver este problema por separado en cada departamento donde, de momento, hay un único encargado que atiende personalmente a cada uno de los clientes. Han realizado cálculos de cuantos clientes van a tener y cuanto tiempo invertirán cada uno de ellos de antemano; es decir, se dispone del dato de que el cliente i requerirá un tiempo t_i para $1 \leq i \leq n$.

Hay que tener en cuenta que el tiempo que invierte cada cliente se cuenta desde su entrada en la tienda hasta su salida. Como las ofertas son muy buenas, todos los clientes estarán en la puerta a la hora de la apertura, y el encargado irá pidiendo que pasen en el orden dado por el algoritmo que se te pide construir, por lo que tendrán que esperar a que el encargado atienda a los clientes que van delante de él.

Se pide resolver este problema mediante una técnica voraz (devorador).

- a) Plantear un heurístico para la función de selección y demostrar que es el mejor frente a otros, suponiendo que tenemos previstos tres clientes, con $t_1= 8$, $t_2= 15$ y $t_3= 5$.
- b) Escribir en Java el algoritmo para dar solución a este problema.

7. (0,5 puntos) Razona brevemente por qué el algoritmo de ramificación y poda utilizamos como estructura de datos de apoyo una *cola de prioridad* frente a otras estructuras de datos secuenciales.