

EXAMEN DE METODOLOGÍA DE LA PROGRAMACIÓN – ALGORITMIA

Primer parcial – Febrero 99 (Turno de mañana)

- Definición matemática de algoritmo. Diferencias con el concepto de cálculo.
- Formalizar mediante un método de cálculo el algoritmo de búsqueda binaria. La salida debe indicar si el elemento se encontró. En caso afirmativo, indicará la posición del elemento. Comprobar con ejemplos la corrección del método. Usar la definición :

Vector = $(a_1, a_2, \dots, a_n)$ ($a_i \in \mathbb{Z}$ y $n \in \mathbb{Z}^+$)

Tam(vector) = n

Elem(vector, i) = a_i $i \in 1 \dots \text{tam}(\text{vector})$

- Definir una máquina de Turing que obtenga el and lógico de dos números binarios separados por el símbolo x.
- La ordenación por inserción puede formularse de manera recursiva como : para ordenar el array $A[1 \dots n]$, se ordena recursivamente el array $A[1 \dots n-1]$ y después se inserta $A[n]$ ordenadamente.
 - Escribir una versión recursiva y analizar por todos los métodos que conozca el tiempo de ejecución exacto en el peor caso, mejor caso y caso medio.
 - Comparar su eficiencia con la ordenación por selección iterativa y con el merge-sort.
- Diseñar un algoritmo divide y vence que determine si dos árboles son isomorfos. (Dos árboles son isomorfos si ambos son NULL o sus subárboles tienen estructura isomorfa (Izquierda-Izquierda y Derecha-Derecha o bien Izquierda-Derecha y Derecha-Izquierda)).
 - Escribir el pseudocódigo.
 - Analizar el algoritmo.
 - Basándose en lo anterior, diseñar un algoritmo que compruebe si son holomorfos y si tienen el mismo contenido, y analizarlo.

EXAMEN DE METODOLOGÍA DE LA PROGRAMACIÓN – ALGORITMIA

Primer parcial – Junio 99 (Turno de mañana)

- Desarrollar un método de cálculo para un algoritmo que devuelve el mínimo y el máximo de los elementos de una matriz, y las posiciones donde se encuentran.
- Especificar una máquina de Turing que actúe sobre una cadena binaria en una cinta, con el bit más significativo a la izquierda, y que escriba un duplicado exacto de la misma a la derecha de la cadena original.
- Escribir el pseudocódigo de la ordenación por inserción de un array $A[1..n]$ y desarrollar detalladamente el análisis del número de comparaciones del bucle interno en el peor caso y en el caso medio.
- Sea T un vector de n elementos, intercambiar los k primeros elementos del array y los n-k últimos sin utilizar un vector auxiliar. Ej :

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11) $k = 3$

(4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3)

- Diseñar un algoritmo divide y vence para resolverlo
- Analizar el tiempo de ejecución del algoritmo, expresándolo en función del número de intercambios de elementos del array

EXAMEN DE METODOLOGÍA DE LA PROGRAMACIÓN – ALGORITMIA

Segundo parcial – Febrero 99 (Turno de mañana)

- Verificar la corrección parcial de un fragmento de programa que efectúe una búsqueda lineal de un elemento x en un array $T[1..n]$ comenzando el recorrido en la posición n .
- Diseñar un algoritmo probabilista para comprobar si dos cadenas de caracteres son iguales. Explicar que tipo de algoritmo es y sus características fundamentales. Analizarlo y compararlo con la versión determinista, y decidir cuál es mejor.
- Tenemos un grafo conexo no dirigido y valorado (positivo) $G = \{V, E\}$. Llamaremos P a uno cualquiera de sus caminos. Se define $\text{PesoMáximo}(P)$ como el máximo de todos los pesos de los arcos de P . Desarrollar, demostrar la corrección y analizar el tiempo de ejecución de un algoritmo en tiempo polinomial para el problema : dados dos vértices cualesquiera del grafo G , s y t , encontrar uno de los caminos entre ambos nodos que minimice la función $\text{PesoMáximo}(P)$.
- Tenemos un editor de textos con una función para transformar una cadena de caracteres fuente en otra destino. Internamente el editor sólo puede realizar :
 - Eliminar el primer carácter del fuente
 - Añadir al final del destino un caracter
 - Copiar el primer caracter fuente al último de destino, eliminando el fuente
 - Reemplazar

Cada operación tendrá un coste positivo $C[i]$. El coste total será la suma de los elementales.

Se pide :

- Diseñar y analizar (en tiempo y en espacio) un algoritmo de programación dinámica que encuentre una de las transformaciones de coste mínimo para una cadena fuente y otra destino $D = (d_1, d_2, \dots, d_n)$



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

Facultad de Informática

