

Algoritmia. Curso 2011/12

Orientaciones de la práctica

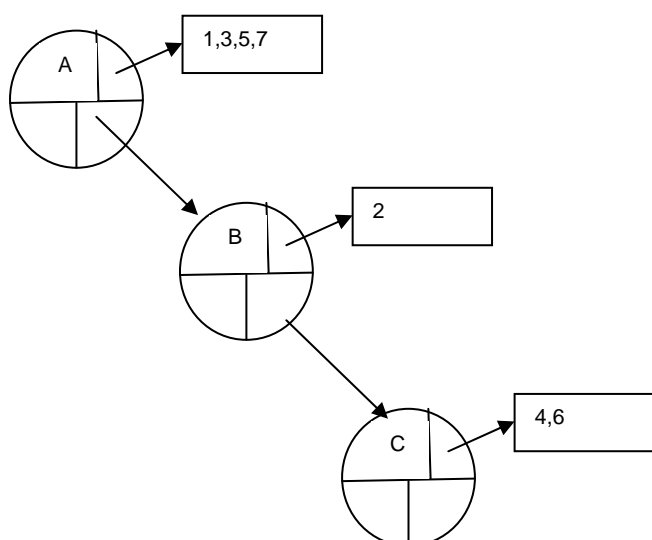
Sea la estructura de datos principal A que relaciona los datos de los estudiantes (nombre, dni, edad, nota y los árboles de índices B y C, donde se mantienen ordenadas las referencias a los datos de los estudiantes siguiendo el orden alfabético de sus nombres y el orden creciente de sus edades, respectivamente.

Un ejemplo de dichas estructuras se muestra a continuación:

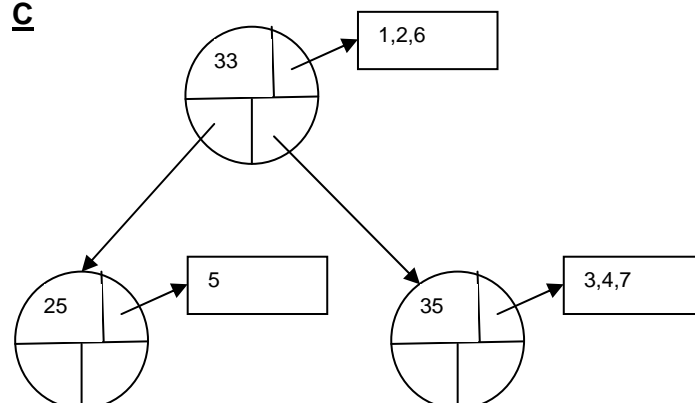
A

Id	Nombre	DNI	Edad	Nota media
1	Antonio		33	8
2	Bernardo		33	9
3	Andrés		35	5
4	Concha		35	6
5	Alejandro		25	8
6	Carlos		33	6
7	Amparo		35	6
	...			

B



C



Ejercicios

Utilizando las estructuras de datos (listas, pilas, colas, árboles, etc.), y los métodos que las gestionan, implementados durante las prácticas de clases del tema 2:

(1.5 punto) Implementar las estructuras de datos que se definen anteriormente.

(2.5 puntos) Implementar las operaciones básicas de inicialización, inserción, eliminación, etc. (Hay que tener en cuenta que al operar con la lista A también se debe operar con los árboles B y C.)

Las operaciones básicas deben estar implementadas en métodos independientes, siguiendo buenas prácticas de programación modular.

- void **inicializar** (Lista A, ArbolIndice B, ArbolIndice C) – inicializar la lista A y los árboles de índices B y C.
- void **insertar** (Lista A, ArbolIndice B, ArbolIndice C, char[] nombre, char[] dni, int edad, int nota) – inserta el estudiante especificado en la lista A y actualiza los árboles de índices B y C.
La inserción en las estructuras de índice puede realizarse de manera ordenada o siempre por la cabeza de la estructura de índices bajo el nodo del árbol en cuestión.
- int **eliminar** (Lista A, ArbolIndice B, ArbolIndice C, char[] dni) – elimina el estudiante especificado de la lista A y actualiza los árboles de índices B y C. Devuelve -1 si el estudiante no se encuentra en la estructura.

(2.5 puntos) Implementar las siguientes operaciones:

- void **imprimir_letra_dada**(Lista A, ArbolIndice B, char letra) – imprime en orden alfabético todos los datos de todos los estudiantes cuya letra inicial sea letra.
- void **imprimir_orden_alfabetico**(Lista A, ArbolIndice B) – imprime en orden alfabético todos los datos de todos los estudiantes.
- void **imprimir_edad_dada**(Lista A, ArbolIndice C, int edad) – imprime en orden alfabético todos los datos de todos los estudiantes cuya edad sea edad.
- void **imprimir_orden_edad**(Lista A, ArbolIndice C) – imprime por edades, todos los datos de todos los estudiantes.

(2.5 puntos) Estudio comparativo sobre la eficiencia de tres alternativas para ofrecer los listados ordenados de los estudiantes por edad (método **imprimir_orden_edad**). Las variantes pueden venir dadas por la forma en que se insertan y/o recuperan los índices bajo el nodo del árbol que corresponde a una determinada edad:

Estrategia 1. Insertar, sobre la lista de índices de cada edad, de manera ordenada (por lo tanto, la recuperación se produce en orden).

Estrategia 2. Insertar, sobre la lista de índices de cada edad, sin seguir ningún criterio de orden. Por lo tanto, la lista de índices de una determinada edad debe ser ordenada, **mediante el método de la burbuja**, en momento de recuperar (e imprimir) los datos los estudiantes.

Estrategia 3. Similar a la estrategia 2 pero utilizando el método de **ordenación por mezcla**.

Para comparar la eficiencia puede utilizarse como criterio el tiempo consumido en la ejecución de cada variante. (Este tiempo puede estimarse mediante la diferencia entre el instante de tiempo en que comienza la ejecución de un determinado método y el instante en que finaliza.)

Es necesario utilizar un número relativamente alto de estudiantes con la misma edad (200 – 500) de manera que sea posible identificar a partir de qué cantidad de estudiantes es conveniente utilizar las estrategias 2 ó 3 en lugar de la 1.

Se debe generar un documento Word donde se ofrezcan gráficas y juicios comparativos sobre dicho análisis.

(1.0 puntos) Calidad de la implementación y del estudio de eficiencia realizado.

Entregables

Mediante la tarea correspondiente en el campus virtual, el día 04/06/12 09:00am:

- 1) Un único fichero .C donde se presenten de manera progresiva, separada y con los comentarios oportunos todas las definiciones de tipo, constantes y métodos utilizados.
De manera obligada se debe incluir la función **main** y las funciones independientes solicitadas en los diferentes apartados de la práctica.
- 2) Un documento Word donde se compare la eficiencia de cada una de las variantes analizadas.