

Universidad Nacional Andrés Bello

Estructuras de Datos — Solemne 3

Profesores: Luis Bastías, Rodrigo Paredes, Iván Zuñiga
Ayudantes: Patricia Albornoz, Francisco Claude, Hans Ulloa
Fecha: 23 de Noviembre de 2006
Duración 1h y 30m

1. Hash:

Se desea reemplazar una tabla de hash abierto (hash implementado con listas enlazadas) de B1 celdas con N elementos (N bastante mayor que B1), por otra tabla de hash cerrado de B2 celdas ($N/B2=0,5$). Genere un algoritmo, tipo C o JAVA, para llevar a cabo esta tarea.

Solución:

```
Crear Nueva Tabla;
Para cada lista de la tabla de hash abierto {
  Para cada elemento de la lista {
    Sacar elemento de la lista
    Insertar elemento en hash cerrado (rehash lineal)
  }
}
```

2. Ordenamiento:

Modifique el algoritmo de ordenamiento quicksort de forma tal que cuando los subarreglos, generados en el proceso de particion, tengan un tamaño igual o menor a una constante M, continúe el ordenamiento con algún algoritmo de orden cuadrático visto en clases.

```
QuickSort(A, ini, fin)
  if (fin - ini < M) return // la partición es chica, corto
  pivotId = floor((ini+fin)/2) // tomo el pivote
  pivotId = particiona(A, ini, fin, pivotId) // particiono
  QuickSort(A, ini, pivotId - 1) // recursion izquierda
  QuickSort(A, pivotId + 1, fin) // recursion derecha
```

Metodo Principal:

```
Sort(A, sizeA)
  QuickSort(A, 0, sizeA - 1) // orden parcial del arreglo
  InsertSort(A, sizeA - 1) // pasada final de InsertSort.
```

```
Llamada inicial:  
Sort(A, lenght(A))
```

3. Arboles:

Para un árbol binario escriba un algoritmo, tipo C o Java, que pueda escribir sólo los nodos internos (sin considerar las hojas) y además los escriba desde la izquierda a la derecha.

Solucion:

```
Escribir(TREE *T) {  
Si T != NULL {  
    Si (T->izq != NULL)  
        Escribir(T->izq)  
    Si (T->izq != NULL) o (T->der != NULL)  
        Imprimir T->elemento  
    Si (T->der != NULL)  
        Escribir(T->der)  
}
```

Otra solucion mas corta:

```
Escribir(Tree *T)  
    if (T == NULL) return // arbol vacio, nada que hacer  
    internos(T.izq) // llamada recursiva a la izquierda  
    if (T.izq != T.der) print(T.info) // imprimo el nodo no hoja  
    internos(T.der) // llamada recursiva a la derecha
```

llamada inicial: Escribir(arbol)