

Examen de Programación 2

Diciembre de 2022

Problema 1 (50 puntos)

Se quiere representar una estructura de directorios (carpetas) de un sistema operativo, donde cada directorio tiene un nombre único que lo identifica. Cada directorio puede contener un número finito de subdirectorios y una lista de archivos, donde cada archivo tiene un nombre y un contenido.

Considere el tipo **Directorios**, definido como árboles generales de *Archivos* e implementado como árboles binarios con la semántica primer hijo (pH) – siguiente hermano (sH):

<pre>struct nodoArchivo { char * nombreArchivo; char * contenidoArchivo; nodoArchivo * sig; }; typedef nodoArchivo * Archivos;</pre>	<pre>struct nodoDirectorio { char * nombreDirectorio; Archivos listaArchivos; nodoDirectorio * pH, * sH; }; typedef nodoDirectorio * Directorios;</pre>
--	---

Defina un procedimiento **borrar**

void borrar (Directorios & D, char * nom_dir)

que, dados un directorio D sin elementos repetidos y el nombre nom_dir de un directorio, elimine a nom_dir de D si nom_dir está en D . En caso contrario, el procedimiento no tendrá efecto. Al eliminar un directorio deberá liberarse toda la memoria asociada a éste, incluyendo la correspondiente a los archivos que contenga y a sus subdirectorios. Utilice la función `strEq` (que se asume implementada) para comparar strings; `strEq` retorna `true` si y sólo si dos strings son iguales. Si usa funciones o procedimientos auxiliares, deberá implementarlos.

Problema 2 (50 puntos)

- a) Especifique el TAD **Tabla** (mapping, función parcial) no acotada con dominio y rango de tipo *int*. Considere operaciones para: 1) crear una tabla vacía, 2) agregar una correspondencia, 3) consultar la cantidad de correspondencias, 4) saber si un valor del dominio tiene imagen, 5) obtener la imagen de un valor del dominio, 6) eliminar una correspondencia, y 7) destruir una tabla.
- b) Implemente la función **composicionRestringida**

Tabla composicionRestringida (Tabla t1, Tabla t2, int min, int max)

que, dadas dos tablas $t1$ y $t2$ (de tipo Tabla) y dados dos enteros min y max (con $min < max$), retorne una tabla $t3$ que represente la composición de las funciones parciales representadas por $t1$ y $t2$, con valores en el dominio de $t1$ exclusivamente en el rango $[min, max]$. Esto es, para cada valor entero d en $t1$, tal que $min \leq d \leq max$, $t3(d)$ debería ser $t2(t1(d))$. En caso que $t2$ no esté definida para $t1(d)$, d no debería aparecer en el dominio de $t3$. Implemente la función **composicionRestringida** considerado la **especificación** (y **no** una implementación) del TAD Tabla.

Ejemplo: Sean $t1 = \{(2 \rightarrow 10), (7 \rightarrow 120), (3 \rightarrow 8), (1 \rightarrow 22), (101 \rightarrow 4)\}$ y $t2 = \{(120 \rightarrow 77), (4 \rightarrow 54), (8 \rightarrow 38)\}$, $composicionRestringida(t1, t2, 0, 99) = \{(7 \rightarrow 77), (3 \rightarrow 38)\}$.

- c) Asumiendo una implementación del TAD *Tabla* usando *árboles binarios de búsqueda*, ¿cuál sería el orden de tiempo de ejecución en el caso promedio de la operación **composicionRestringida**? Explique brevemente.

Examen de Programación 2

Diciembre de 2022

SOLUCIONES

PROBLEMA 1

```
void borrar (Directorios & D, char * nom_dir) {
    if (D != NULL) {
        if (strEq(D->nombreDirectorio, nom_dir)) {
            Directorios aBorrar = D;
            D = D->sH;
            borrarTodo(aBorrar->pH);
            delete [] borrar->nombreDirectorio;
            borrarArchivos(borrar->listaArchivos);
            delete aBorrar;
        } else {
            borrar(D->sH, nom_dir);
            borrar(D->pH, nom_dir);
        }
    }
}
```

```
void borrarArchivos (Archivos & A) {
    if (A != NULL) {
        borrarArchivos(A->sig);
        delete [] A->nombreArchivo;
        delete [] A->contenidoArchivo;
        delete A;
        A = NULL;
    }
}
```

```
void borrarTodo (Directorios & D) {
    if (D != NULL) {
        borrarTodo(D->sH);
        borrarTodo(D->pH);
        delete [] borrar->nombreDirectorio;
        borrarArchivos(borrar->listaArchivos);
        delete D;
        D = NULL;
    }
}
```

PROBLEMA 2

```
a)
    struct RepresentacionTabla;
    typedef RepresentacionTabla * Tabla;
    typedef int Dominio;
    typedef int Rango;
```

Examen de Programación 2

Diciembre de 2022

```
// POS: Devuelve la Tabla vacía.
Tabla crearTabla ();

/* POS: Agrega la correspondencia (d,r) en t, si d no tenía imagen en t. En caso
contrario actualiza la imagen de d con r. */
void insertarTabla (Dominio d, Rango r, Tabla & t);

// POS: Devuelve la cantidad de correspondencias en t.
int cantidadTabla (Tabla t);
// Alternativamente podría ponerse el predicado es tabla vacía.

// POS: Devuelve true si y sólo si d tiene imagen en t.
bool estaDefinidaTabla (Dominio d, Tabla t);

// PRE: estaDefinidaTabla(d,t). POS: Retorna la imagen (el valor del rango) de d en t.
Rango recuperarTabla (Dominio d, Tabla t);

/* POS: Elimina de t la correspondencia que involucra a d, si d está definida en t. En
otro caso la operación no tiene efecto. */
void eliminarTabla (Dominio d, Tabla & t);

// POS: Libera toda la memoria ocupada por t.
void destruirTabla (Tabla & t);
```

b)

```
Tabla composicionRestringida(Tabla t1, Tabla t2, int min, int max){
    Tabla t3 = crearTabla();
    for (int d = min; d <= max; d++){
        if (estaDefinidaTabla(d,t1))
            int imagenT1 = recuperarTabla(d,t1);
            if (estaDefinidaTabla(imagenT1,t2))
                insertarTabla(d,recuperarTabla(imagenT1,t2),t3);
    }
    return t3;
}
```

c)

Sería $O((max-min) \cdot \log_2(n))$ promedio, siendo n la cantidad de correspondencias de la tabla, ya que itera en el rango $[min, max]$ realizando acciones de $O(\log_2(n))$ promedio (*estaDefinidaTabla*, *recuperarTabla* e *insertarTabla*). La operación *crearTabla* no aumenta el orden, ya que está en secuencia e insume $O(1)$ peor caso.