AVL FILE & Random File

Organización de archivos

Profesor: Heider Sanchez
ACLs: Josué Arriaga, Mariana Capuñay, Sebastián Loza

P1 (12 pts) AVL File

Usando como base los algoritmos de búsqueda e inserción del AVL, se le pide adaptarlos para trabajar con archivos de registros de longitud fija.

a) Implemente la función **find** para buscar registro dado un search-key.

b) Implemente la función **insert** para agregar un nuevo registro.

c) Implemente una función **inorder** para leer todos los registros ordenados.

d) Implemente una función **remove** para eliminar un registro dado un search-key.

```
struct Record
{
    int codigo;
    char nombre[12];
    char apellido[12];
    int ciclo;
    // Agregar campos extra si los necesitan
};

class AVLFile
{
private:
    string filename;
    long pos_root;
public:
    AVLFile(string filename);
    Record find(TK key);
    void insert(Record record);
    vector<Record> inorder(TK key);
    true remove();
};
```

P2 (8 pts) Random File

Usando como base la implementación de archivos con registros de longitud fija y la estructura Registro, se le pide implementar lo siguiente:

a) Construir un índice (diccionario) para mantener asociado el search-key con las ubicaciones de los registros en el archivo de datos. El índice debe ser cargado a memoria principal al iniciar el programa.

b) Implementar la función **find** para buscar un registro dado un search-key.

c) Implemente la función **insert** para agregar un nuevo registro.

d) Implemente la función **scanAllByIndex** para mostrar todos los registros ordenados en función del índice.

e) Implemente la función **remove** para eliminar un registro usando alguna estrategia de eliminación eficiente.

f) Después de cada actualización del índice, éste debe guardarse en memoria secundaria. Cuando finalice el programa, también debe regresar el índice al disco.

```
class RandomFile
{
private:
    string fileName;
    string indexName;
    // Agregar campo extra  para mantener el indice en memoria principal

public:
    RandomFile(string _fileName){
        ...
        index = leerIndice(); //desde disco duro
        ...
    }

    ~RandomFile(){
        ...
        guardarIndice(index);
        delete index;
        ...
    }
};
```

Entregable:

- Suba a Canvas el código de la solución compilable (`p1.cpp`, `p2.cpp`)
- Se debe incluir las pruebas de funcionalidad de cada operación en el `main`.