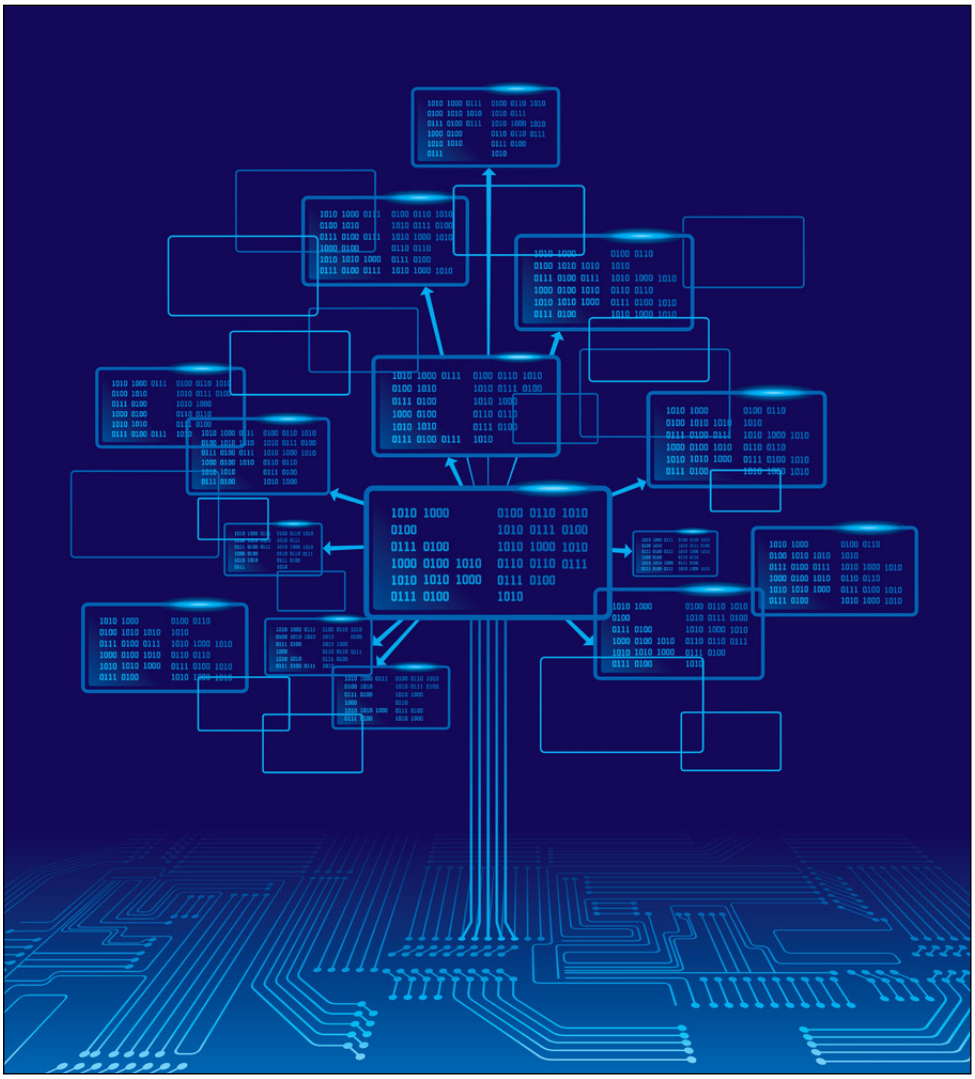
Binary search tree

Reporte 2

DIVISIÓN DE INGENIERÍAS CAMPUS IRAPUATO-SALAMANCA
Algoritmos y estructuras de datos II
Dr. Carlos Hugo García Capulín
Búsqueda de árbol binaria
Alejandro Alonso Sánchez 17 May 2024



INTRODUCCIÓN

This report presents a Binary Search Tree (BST) implementation designed to manage UDA (unidad de aprendizaje) data for the major in engineering for computer science (LISC) of the DICIS. The code offers functionalities for storing course information, searching for courses by key (Clave UDA) and name (Nombre UDA), and deleting courses from the BST. The data structure takes advantage of the efficient search capabilities of BSTs, enabling quick retrieval of specific courses based on their unique key or name.

DESARROLLO

Here's a breakdown of the key functionalities with code snippets:

Data Structure and File Handling:

- REGISTROS Structure: This structure defines the format of a course record, including fields like key (Clave UDA), name (Nombre UDA), type, area, credits, and hours.

```c
typedef struct{
    char ClaveUDA[10];
    char NombreUDA[60];
    char TipoUDA[80];
    char AreaUDA[80];
    unsigned int NumCred;
    unsigned int Hrs;

} REGISTROS;
```

Code 1. REGISTROS structure code

GetRegistro Function:

This function retrieves a specific course record from a file based on its index. It opens the file, skips to the desired line, and reads the data into a REGISTROS structure.

```c
REGISTROS* GetRegistro(int indice, char filename[]) {
    FILE* file = fopen(filename, "r");
    if (file == NULL) {
        printf("Error opening file: %s\n", filename);
        return NULL;
    }

    REGISTROS* registro = malloc(sizeof(REGISTROS));
    if (registro == NULL) {
        printf("Error allocating memory for registro.\n");
        fclose(file);
        return NULL;
    }
    for (int i = 0; i < indice + 1; i++) {
        if (fgetc(file) == EOF) {
            free(registro);
            fclose(file);
            return NULL;
        }
        while (fgetc(file) != '\n') {}  // Read until the end of the line
    }

    fscanf(file, "%s %s %s %s %d %d", registro->ClaveUDA, registro->NombreUDA, registro->TipoUDA, registro-
>AreaUDA, &registro->NumCred, &registro->Hrs);

    fclose(file);
    return registro;
}
```

Code 2.Retrieve registro information