



Universidad Autónoma de Aguascalientes

Estructuras Computacionales

INGENIERÍA EN COMPUTACIÓN INTELIGENTE 2° A

Proyecto 5 búsqueda parcial II

Nombre de los alumnos:

Ángel David Ortiz Quiroz ID: 261481

Ximena Rivera Delgadillo ID: 261261

Erick Iván Ramírez Reyes ID: 260806

Diego Emanuel Saucedo Ortega ID: 261230

José Luis Sandoval Pérez ID: 261731

Carlos Daniel Torres Macias ID: 244543

Profesor: Miguel Ángel Meza de Luna

Fecha de entrega: 03 de junio del 2022

Método Secuencial, con listas y binario

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <stdio.h>

using namespace std;

struct Nodo {
    int dato;
    Nodo* siguiente;
};

void menu1();
void inserLista(Nodo*&, int);
void buscarLista(Nodo*, int);
void insertarVector(int);
void buscarVectorLineal(int);
void buscarVectorBinario(int);
Nodo* lista = NULL;

int vector[20];
int main()
{
    for(int i=0; i<20; i++)
    {
        vector[i] = 0;
    }
    menu1();
    return 0;
}

void menu1()
{
    int op, dato;

    do{
        cout << "\t.:MENU:." << endl;
        cout << "1. Insertar elementos a la lista" << endl;
        cout << "2. Buscar un elemento en la lista de manera secuencial" << endl;
        cout << "3. Insertar elemento al vector" << endl;
        cout << "4. Buscar un elemento en el vector de manera secuencial" << endl;
        cout << "5. Buscar un elemento en el vector de manera binaria" << endl;
        cout << "6. Salir" << endl;
        cin >> op;
        switch (op) {
            case 1: cout << "Digite un numero: "; cin >> dato; inserLista(lista,
dato); system("pause"); break;

            case 2: cout << "Digite el numero que va a buscar: "; cin >> dato;
buscarLista(lista, dato); cout << "\n"; system("pause"); break;

            case 3: cout << "Digite el numero: "; cin >> dato; insertarVector(dato);
system("pause"); break;
```

```

        case 4: cout << "Digite el numero que va a buscar: "; cin >> dato;
        buscarVectorLineal(dato); cout << "\n"; system("pause"); break;

        case 5: cout << "Digite el numero que va a buscar: "; cin >> dato;
        buscarVectorBinario(dato); cout << "\n"; system("pause"); break;

        case 6: cout << "Adios..."; system("pause"); break;

        default: cout << "Ingrese una opcion valida"; break;

    } system("cls");
} while(op != 6);
}

void inserLista(Nodo*& lista, int n)
{
    Nodo* nuevo_nodo = new Nodo();

    nuevo_nodo->dato = n;

    Nodo* aux1 = lista;

    Nodo* aux2 = NULL;

    while ((aux1 != NULL) && (aux1->dato < n))
    {
        aux2 = aux1;
        aux1 = aux1->siguiente;
    }
    if (lista == aux1)
    {
        lista = nuevo_nodo;
    }
    else
    {
        aux2->siguiente = nuevo_nodo;
    }
    nuevo_nodo->siguiente = aux1;

    cout << "Elemento " << n << " insertado correctamente a la lista\n";
}

void buscarLista(Nodo* lista, int n)
{
    bool band = false;

    Nodo* actual = new Nodo();
    actual = lista;

    while ((actual != NULL) && (actual->dato <= n))
    {
        if (actual->dato == n)
        {
            band = true;
        }
        actual = actual->siguiente;
    }
}

```

```

    }

    if (band == true)
    {
        cout << "Elemento " << n << " SI ha sido registrado en la lista\n";
    }
    else {
        cout << "Elemento " << n << " NO ha sido registrado en la lista\n";
    }
}

void insertarVector(int n)
{
    bool x = false;
    int i = 0;

    if (vector[20] != 0)
    {
        cout << "Vector lleno, reinicie" << endl;
    }
    else
    {
        do {
            for(i = 0; i < 20; i++)
            {
                if (vector[i] == 0 && x == false)
                {
                    vector[i] = n;
                    x = true;
                }
            } while (i < 20 || x == false);
        } while (i < 20 || x == false);
        cout << "Elemento " << n << " insertado correctamente a la lista\n";
    }
}

void buscarVectorLineal(int n){

    bool band = false;

    int p = 0;

    for(int i = 0; i < 20; i++){
        if (vector[i] == n){
            band = true;
            p = i+1;
        }
    }

    if (band == true)
    {
        cout << "Elemento " << n << " SI ha sido registrado en el vector, en la
posicion " << p << endl;
    }
    else
    {

```

```

        cout << "Elemento " << n << " NO ha sido registrado en el vector, en la
posicion " << p << endl;
    }
}

void buscarVectorBinario(int n)
{
    int vector2[20];
    int mitad, i, j, valorC, inicio = 0, fin = 19;
    bool band = false;
    for (i = 0; i<20; i++)
    {
        vector2[i] = vector[i];
    }

    for (i = 0; i<20; i++)
    {
        for(j = 0; j<20; j++)
        {
            if (vector2[i]<vector2[j]){
                swap (vector2[i], vector2[j]);
            }
        }
    }

    while(inicio <= fin){
        mitad = ((inicio + fin)/2);
        valorC = vector2[mitad];

        if (!band && n == valorC)
        {
            band = true;
        }

        else if(n < valorC)
        {
            fin = mitad - 1;
        }
        else
        {
            inicio = mitad + 1;
        }
    }

    if (band == true)
    {
        cout << "Elemento " << n << " SI ha sido registrado en el vector" << endl;
    }
    else
    {
        cout << "Elemento " << n << " NO ha sido registrado en el vector" << endl;
    }
}

```

Comparativa

Método Secuencial

El muestreo secuencial es una técnica de muestreo no probabilístico en donde el investigador escoge un sujeto o un grupo de sujetos en un determinado intervalo de tiempo, lleva a cabo su estudio, analiza los resultados, luego escoge otro grupo de sujetos, si es necesario, y así sucesivamente.

Si vamos a tener en cuenta todas las demás técnicas de muestreo en la investigación, llegaremos a la conclusión de que el experimento y el análisis de datos se resumirán en la aceptación de la hipótesis nula o la refutación de ella en caso de aceptar la hipótesis alternativa.

Esta técnica de muestreo brinda al investigador posibilidades ilimitadas de ajustar sus métodos de investigación y obtener un conocimiento fundamental sobre el estudio que está llevando a cabo.

En la técnica de muestreo secuencial existe un paso más, una tercera opción. El investigador puede aceptar la hipótesis nula, aceptar su hipótesis alternativa o seleccionar otro grupo de sujetos y llevar a cabo el experimento nuevamente. Esto significa que el investigador puede obtener un número ilimitado de sujetos antes de tomar una decisión final sobre si acepta la hipótesis nula o la alternativa.

Ventajas del muestreo secuencial

El investigador tiene una opción ilimitada con respecto al tamaño de la muestra y al programa de muestreo. El tamaño de la muestra puede ser relativamente pequeño o excesivamente grande dependiendo de la decisión del investigador. El programa de muestreo también depende completamente del investigador, ya que un segundo grupo de muestras sólo puede obtenerse luego de llevar a cabo el experimento con el primer grupo.

Como mencionamos anteriormente, esta técnica de muestreo permite que el investigador ajuste sus métodos de investigación y análisis de resultados. Debido a la naturaleza repetitiva de este método de muestreo, se pueden realizar pequeños cambios y ajustes durante la parte inicial del estudio para corregir y perfeccionar el método de investigación.

Hay muy poco esfuerzo por parte del investigador en esta técnica de muestreo. No es cara, no consume mucho tiempo ni necesita mucha mano de obra.

Desventajas del muestreo secuencial

Este método de muestreo difícilmente sea representativo de toda la población. Su única esperanza de lograr representatividad es cuando el investigador elige utilizar una muestra lo suficientemente grande como para representar una gran fracción de toda la población.

La técnica de muestreo también difícilmente sea aleatoria. Esto contribuye al poco grado de representatividad de la técnica de muestreo.

Debido a las desventajas anteriormente mencionadas, los resultados de esta técnica de muestreo no pueden ser utilizados para desarrollar conclusiones e interpretaciones con respecto a toda la población.

Método Binario

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento en una lista ordenada de elementos. Funciona al dividir repetidamente a la mitad la porción de la lista que podría contener al elemento, hasta reducir las ubicaciones posibles a solo una.

Al describir un algoritmo para un ser humano, a menudo es suficiente una descripción incompleta. Algunos detalles pueden quedar fuera de una receta de un pastel; la receta supone que sabes cómo abrir el refrigerador para sacar los huevos y que sabes cómo romper los huevos. La gente puede saber intuitivamente cómo completar los detalles faltantes, pero los programas de computadora no. Es por eso que necesitamos describir completamente los algoritmos computacionales.

Veamos cómo describir cuidadosamente la búsqueda binaria. La idea principal de la búsqueda binaria es llevar un registro del rango actual de intentos razonables.

La búsqueda binaria al igual que otros algoritmos como el quicksort utiliza la técnica divide y vencerás. Uno de los requisitos antes de ejecutar la búsqueda binaria, es que el conjunto de elementos debe de estar ordenado. Supongamos que tenemos el siguiente array.