



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO



REPORTE DE EXAMEN 2

NOMBRE DEL ALUMNO: GARCÍA QUIROZ GUSTAVO IVAN

GRUPO: 2CV3

MATERIA: ALGORITMOS Y ESTRUCTURA DE DATOS

FECHA: 09/12/2022

Índice

Introducción	3
Desarrollo	4
Resultados	7
Conclusiones	8
Bibliografía	9

Introducción

Las estructuras de datos vistas como las tablas hash y las pila ayuda a mejorar el algoritmo ya que a partir de los datos de un registro se calcula, en primer lugar, un valor hash. Los valores hash de todos los registros de una base de datos se guardan en la tabla hash. Mediante otra operación matemática, a partir del valor hash se calcula la ubicación de dicha información en la base de datos. Si el usuario introduce entonces un término en el campo de búsqueda, este término también se hashea.

En cuanto a los puntos débiles de este método, uno de ellos es la posible degeneración grave del sistema si se producen muchas colisiones. La probabilidad de las colisiones aumenta a medida que lo hace la cantidad de datos almacenados. La presencia de una gran cantidad de funciones hash impide movimientos de un conjunto de datos al anterior o al siguiente.

Las pilas por otra parte es un conjunto ordenado de elementos en el cual se pueden agregar y eliminar elementos de un extremo, el cual es llamado el tope de la pila.

La pila es un objeto dinámico en constante cambio. La definición de pila especifica que un solo extremo de la pila se designa como tope.

Pueden colocarse nuevos elementos en el tope de la pila o se pueden quitar elementos de él.

La característica más importante de la pila es que el último elemento insertado en ella es el primero en suprimirse.

Por esta razón, una pila se denomina como una estructura LIFO (Last In First Out) en la cual, el último elemento insertado, será el primero en ser eliminado.

Desarrollo

En las **tablas hash**, almacena datos en forma de pares de clave y valor. La clave, que se utiliza para identificar los datos, se proporciona como entrada a la función hash. El código hash, que es un número entero, se asigna al tamaño fijo que tenemos.

Las tablas hash deben admitir 3 funciones.

- insertar (clave, valor)
- obtener (la clave)
- borrar (clave)

Simplemente como un ejemplo para ayudarnos a comprender el concepto, supongamos que queremos asignar una lista de claves de cadena a valores de cadena (por ejemplo, asignar una lista de países a sus ciudades capitales).

Y supongamos que nuestra **función hash** es simplemente tomar la longitud de la cadena.

Para simplificar, tendremos dos matrices: una para nuestras claves y otra para los valores.

Entonces, para colocar un elemento en la tabla hash, calculamos su código hash (en este caso, simplemente contamos el número de caracteres), luego colocamos la clave y el valor en las matrices en el índice correspondiente.

En el código se usa primero la inicialización de variable enteras para almacenar los valores del archivo datos.txt.

```
int main(int argc, char *argv[]) {
// int Datos[funcion][2]={69,"Pepe"},{30,"Juan"},{24,"Peralta"},{14,"Ochoa"},{34,"Pena"};
int Datos[75][5]; // unidad, prec, dia, mes, a, status
int hash[75][6]; // unidad, prec, dia, mes, a, status
int funcion=75; // n datos
int resultado=0, i=0, j=0, ele=0;
// almace Datos
int unidad, dia, mes, anio;
int precio;
// imprime Datos
// almacena Datos desde data.txt
```

Después se tuvo que abrir el archivo para que se leyera los valores enteros que contenía el archivo y se almacenaran en la un arreglo de datos, y después se tendría de convertir a tabla hash para que e pudiera buscar mas rápido los valore con un índice..

```
FILE *f;
f = fopen("data.txt", "r");
// unidades / precio / caducidad dia.mes.ano
while( fscanf(f, "%i %i %i %i %i", &unidad,&precio, &dia, &mes,&anio) == 5)
{
// printf("unidad: %i precio: %.2f dia:%i / %i / %i \n", unidad, precio, dia, mes, anio);
Datos[i][0]=unidad;
Datos[i][1]=precio; // clave para el modulo
Datos[i][2]=dia;
Datos[i][3]=mes;
Datos[i][4]=anio;
i++;
}
fclose(f);
printf("Datos almacenados\n");
for(i=0; i<75; i++)
```

```
// Generar tabla hash

for(i=0;i<75;i++){
    for(j=0;j<6;j++){
        hash[i][j]=0;
    }
}

for(i=0;i<=74;i++){
    resultado=Datos[i][0]%funcion;
    printf("%d",resultado);
    //printf("%d %d \n", hash[resultado][2], resultado);
    while((hash[resultado][2])!=0){
        fflush(stdin);
        resultado++;
        if(resultado>74){
            resultado=0;
        }
    }
}
```

El ejercicio trataba acerca de una tienda departamental en donde se debía generar un menú donde se escogiera una de las siguientes 3 opciones:

1. Agregar producto vendido: En esta opción se tenía que almacenar en una pila los productos agregados y se sumó el costo total el cual era calculado mediante el costo normal, más el 60% del producto.
2. Consultar: se consultaba en la tabla hash el valor a buscar mediante un índice tal como se muestra en el código.
3. Eliminar: Esta opción debía desapilar un elemento de la pila y restar el costo el valor sacado del registro.

```
*/ while(ele!=-1){

    printf("\t\t\t\t\t tienda departamental\n");
    printf("----Menu-----\n");
    printf("1. Almacenar venta de un producto\n");
    printf("2. Consultar producto\n");
    printf("3. Borrar venta de un producto\n Elige una opcion \n");
    scanf("%d",&ele);
    switch(ele){
        case 1:{

            int VB=0, existe=0;
            printf("Ingresa el identificador del producto (numero del 0-74) \n ");
            scanf("%d",&VB);
            resultado=VB%funcion;
            while(hash[resultado][0]!=VB){
                resultado++;

                if(resultado>74){
                    resultado=0;
                }
            }
        }
    }
}
```

Para la pila se usaron las funciones pop() y push()

```

void push(int valor){ // push(int valor, nodo *&pcima)
    nodo *nod= new (nodo);
    nod->dato=valor;
    nod->siguiente=pcima;
    pcima=nod;
}
void pop(int valorsal){
    //validamos si pcima=NULL
    if(!pcima){
        printf("\t Pila vacia \n");
        //return= "esta vacia"
    }else{
        nodo *aux= new (nodo);
        valorsal=aux->dato;
        printf("El valor %d es eliminado", pcima->dato);
        //pcima->siguiente=aux->siguiente;
        pcima=pcima->siguiente;
        delete aux;
    }
}

```

Resultados

```
Datos almacenados
unidad: 0 precio: 0 Fecha de caducidad:1 / 12 / 2022
unidad: 1 precio: 10 Fecha de caducidad:2 / 12 / 2022
unidad: 2 precio: 20 Fecha de caducidad:3 / 12 / 2022
unidad: 3 precio: 30 Fecha de caducidad:4 / 12 / 2022
unidad: 4 precio: 40 Fecha de caducidad:5 / 12 / 2022
unidad: 5 precio: 50 Fecha de caducidad:6 / 12 / 2022
unidad: 6 precio: 60 Fecha de caducidad:7 / 12 / 2022
unidad: 7 precio: 70 Fecha de caducidad:8 / 12 / 2022
unidad: 8 precio: 80 Fecha de caducidad:9 / 12 / 2022
unidad: 9 precio: 90 Fecha de caducidad:10 / 12 / 2022
unidad: 10 precio: 100 Fecha de caducidad:11 / 12 / 2022
unidad: 11 precio: 110 Fecha de caducidad:12 / 12 / 2022
unidad: 12 precio: 120 Fecha de caducidad:13 / 12 / 2022
unidad: 13 precio: 130 Fecha de caducidad:14 / 12 / 2022
unidad: 14 precio: 140 Fecha de caducidad:15 / 12 / 2022
unidad: 15 precio: 150 Fecha de caducidad:16 / 12 / 2022
unidad: 16 precio: 160 Fecha de caducidad:17 / 12 / 2022
unidad: 17 precio: 170 Fecha de caducidad:18 / 12 / 2022
unidad: 18 precio: 180 Fecha de caducidad:19 / 12 / 2022
unidad: 19 precio: 190 Fecha de caducidad:20 / 12 / 2022
unidad: 20 precio: 200 Fecha de caducidad:21 / 12 / 2022
unidad: 21 precio: 210 Fecha de caducidad:22 / 12 / 2022
unidad: 22 precio: 220 Fecha de caducidad:23 / 12 / 2022
unidad: 23 precio: 230 Fecha de caducidad:24 / 12 / 2022
unidad: 24 precio: 240 Fecha de caducidad:25 / 12 / 2022
unidad: 25 precio: 250 Fecha de caducidad:26 / 12 / 2022
unidad: 26 precio: 260 Fecha de caducidad:27 / 12 / 2022
unidad: 27 precio: 270 Fecha de caducidad:28 / 12 / 2022
unidad: 28 precio: 280 Fecha de caducidad:29 / 12 / 2022

tienda departamental
|----Menu-----
1. Almacenar venta de un producto
2. Consultar producto
3. Borrar venta de un producto
Elige una opcion
1
Ingresa el identificador del producto (numero del 0-74)
1
Ingresando producto con unidades: 1
Costo total : 16

tienda departamental
----Menu-----
1. Almacenar venta de un producto
2. Consultar producto
3. Borrar venta de un producto
Elige una opcion
2
Ingresa el identificador del producto (numero del 0-74)
2
unidad: 2 precio: 20 Fecha de caducidad:1 / 3 / 12

tienda departamental
----Menu-----
1. Almacenar venta de un producto
2. Consultar producto
3. Borrar venta de un producto
Elige una opcion
```

Conclusiones

En conclusión, las estructuras de datos que se vieron como son las pilas y las tablas hash resultan eficientes en computación porque se puede agilizar un trabajo mediante un procesamiento mas optimo con tan solo escribir un algoritmo que calcule el menor numero de operaciones para un trabajo en especial.

Las pilas las podemos encontrar aplicadas por ejemplo en redes sociales, o también en operaciones específicas como lo es un compilador al momento de revisar que cuando compila el código este correctamente escrito.

Bibliografía

Tablas hash: acceso rápido a valores hash desde la base de datos. (s/f). IONOS
Digital Guide. Recuperado el 9 de diciembre de 2022, de
<https://www.ionos.es/digitalguide/servidores/seguridad/tablas-hash/>