

Análisis y Algoritmos

Roberto Charreton Kaplun
Universidad de Artes Digitales

Guadalajara, Jalisco

Email: idv17c.rcharreton@uartesdigitales.edu.mx

Profesor: Efraín Padilla

Julio 04, 2019

1) Hashing Table

Para resolver el problema de las colisiones se empleo el metodo del encadenamiento. Cada espacio de la matriz contiene un enlace a una lista enlazada individualmente que contiene pares con el mismo hash. Los nuevos pares se agregan al final de la lista.

La funcion de insert mueve al grupo que corresponde al índice hash calculado anteriormente e inserta el nuevo nodo al final de la lista. Para insertar un nodo en la tabla hash, se necesita encontrar el índice de hash para la key.

```
CHash::CHash(int V)
{
    this->m_bucket = V;
    m_table = new list<int>[m_bucket];
}

void CHash::insertItem(int key)
{
    int index = hashFunction(key);
    m_table[index].push_back(key);
}

int Array[] =
{
    15, 11, 27, 8, 12,
    12, 5, 56, 23, 26,
    68, 9, 76, 12, 5,
    56, 23, 26, 68, 9,
    76, 1, 23, 45, 5
};

int main()
{
    int size = sizeof(Array) / sizeof(Array[0]);
    int n = 5;
    CHash HashTable(n);
    for (int i = 0; i < size; i++)
        HashTable.insertItem(Array[i]);

    HashTable.displayHash();
    cin.get();
    cin.ignore();
    return 0;
}
```