**Доклад**

**по**

**Алгоритми и структури от данни**

**Тема: Хеш алгоритми**

1. **Хеширащи алгоритми. Хеш таблица**
2. **Имплементация на хеш-таблица**
3. **Източници**

Изготвил: Ваня Ванева 12а

1. **Хеширащи алгоритми. Хеш таблица**

Хеш алгоритмите са методи за генериране на стойност (наречена хеш) от произволен вход, като обикновено този вход е произволна дължина.

Хеш таблицата е структура от данни, която използва хеш функция за да преобразува входни данни в индекс в масив. Това позволява бързо търсене и достъп до данните.

1. **Имплементация на хеш-таблица**

Имплементацията на хеш-таблица представлява процеса на създаване на структура от данни, която позволява бързо съхраняване и търсене на информация, като използва хеш функция за преобразуване на ключовете в индекси в масив. Тази структура е изградена от две основни компоненти: масив и хеш функция.

**Масив**: В основата на хеш-таблицата се намира масив, който се използва за съхранение на информацията. Всяка клетка в масива може да съдържа нула или повече елементи. Ключовете се преобразуват в индекси в този масив чрез хеш функцията.

**Хеш функция**: Хеш функцията е алгоритъм, който приема ключ и връща числен резултат, наречен хеш код. Този хеш код се използва като индекс в масива, където ще бъде съхранена информацията, свързана с ключа. Целта на хеш функцията е да разпредели ключовете равномерно в масива, за да се намали количеството на колизиите (ситуации, когато два или повече ключа имат един и същи хеш код).

Имплементацията на хеш-таблицата включва определяне на размера на масива, създаване на хеш функция, добавяне на методи за добавяне и търсене на елементи и обработка на колизии (например, чрез използване на свързани списъци или други техники).

Така, имплементацията на хеш-таблицата включва създаването на механизми, които осигуряват ефективност и надеждност при съхранение и търсене на данни в структурата, като се вземат предвид важни аспекти като равномерното разпределение на ключовете и обработката на колизии.

**Примерен код:**

**using System;**

**using System.Collections.Generic;**

**namespace HashTable**

**{**

**public class HashTable<TKey, TValue>**

**{**

**private const int Capacity = 1000;**

**private LinkedList<KeyValuePair<TKey, TValue>>[] items;**

**public HashTable()**

**{**

**items = new LinkedList<KeyValuePair<TKey, TValue>>[Capacity];**

**}**

**public void Add(TKey key, TValue value)**

**{**

**int hash = GetHash(key);**

**if (items[hash] == null)**

**items[hash] = new LinkedList<KeyValuePair<TKey, TValue>>();**

**items[hash].AddLast(new KeyValuePair<TKey, TValue>(key, value));**

**}**

**public TValue Get(TKey key)**

**{**

**int hash = GetHash(key);**

**if (items[hash] != null)**

**{**

**foreach (var pair in items[hash])**

**{**

**if (pair.Key.Equals(key))**

**return pair.Value;**

**}**

**}**

**throw new KeyNotFoundException();**

**}**

**private int GetHash(TKey key)**

**{**

**return Math.Abs(key.GetHashCode()) % Capacity;**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**// Create an instance of HashTable**

**HashTable<string, int> hashTable = new HashTable<string, int>();**

**// Add some key-value pairs**

**hashTable.Add("one", 1);**

**hashTable.Add("two", 2);**

**hashTable.Add("three", 3);**

**// Retrieve values using Get method**

**Console.WriteLine("Value for key 'one': " + hashTable.Get("one"));**

**Console.WriteLine("Value for key 'two': " + hashTable.Get("two"));**

**Console.WriteLine("Value for key 'three': " + hashTable.Get("three"));**

**// Trying to retrieve a value for a non-existing key**

**try**

**{**

**Console.WriteLine("Value for key 'four': " + hashTable.Get("four"));**

**}**

**catch (KeyNotFoundException)**

**{**

**Console.WriteLine("Key 'four' not found.");**

**}**

**}**

**}**

**}**

1. **Източници**

<https://pd.daffodilvarsity.edu.bd/course/material/book-430/pdf_content>

<https://adacomputerscience.org/concepts/struct_hash_table?examBoard=all&stage=all>

<https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/tutorial/#:~:text=Hashing%20is%20implemented%20in%20two,quickly%20retrieved%20using%20hashed%20key>.

<https://www.geeksforgeeks.org/hash-table-data-structure/>

<https://youtu.be/FsfRsGFHuv4?si=WBaTFvswInhZi1Yb>