

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики

Кафедра програмування

Лабораторна робота №4
ХЕШ-ТАБЛИЦЯ
з курсу “Алгоритми та структури даних”

Виконав:
Студент групи ПМІ-12
Бенько Володимир Сергійович

Львів – 2023

Хеш-Таблиця

Хеш-Таблиця – нелінійна структура даних, елементи якої зберігаються в вигляді пар ключ-значення та впорядковані певним чином. Таблиця дозволяє на пряму доступатись до елементів, використовуючи відповідні їм ключі. Дана реалізація структури даних Хеш-Таблиця має вигляд масиву бінарних дерев пошуку, які використовуються для вирішення колізій. Вона містить масив об'єктів класу Table, реалізованого бінарним деревом пошуку. Клас Table містить приховану структуру Node, яка містить ключ, значення та посилання на лівого та правого синів заданої вершини, а також методи обходу дерева для очищення та виведення. Також клас HashTable містить лічильник кількості записів та хеш функцію, що використовується для рівномірного розподілення елементів по комітках, деструктор та ряд публічних методів:

- HashTable(s) - конструктор, який приймає кількість комірок "s" які потрібно створити
- Add(k, v) - додає новий рядок у таблицю з ключем "k" та значенням "v".
- Find(k) - знаходить рядок у таблиці з ключем "k" і повертає вказівник на його значення.
- Replace(k, v) - замінює значення рядка у таблиці з ключем "k" на нове значення "v".
- Remove(k) - видаляє рядок у таблиці з ключем "k".
- Clear() – видаляє всі записи з таблиці.
- GetCount() - повертає кількість рядків у таблиці.
- Output(ostream out) – виводить таблицю в потік out.

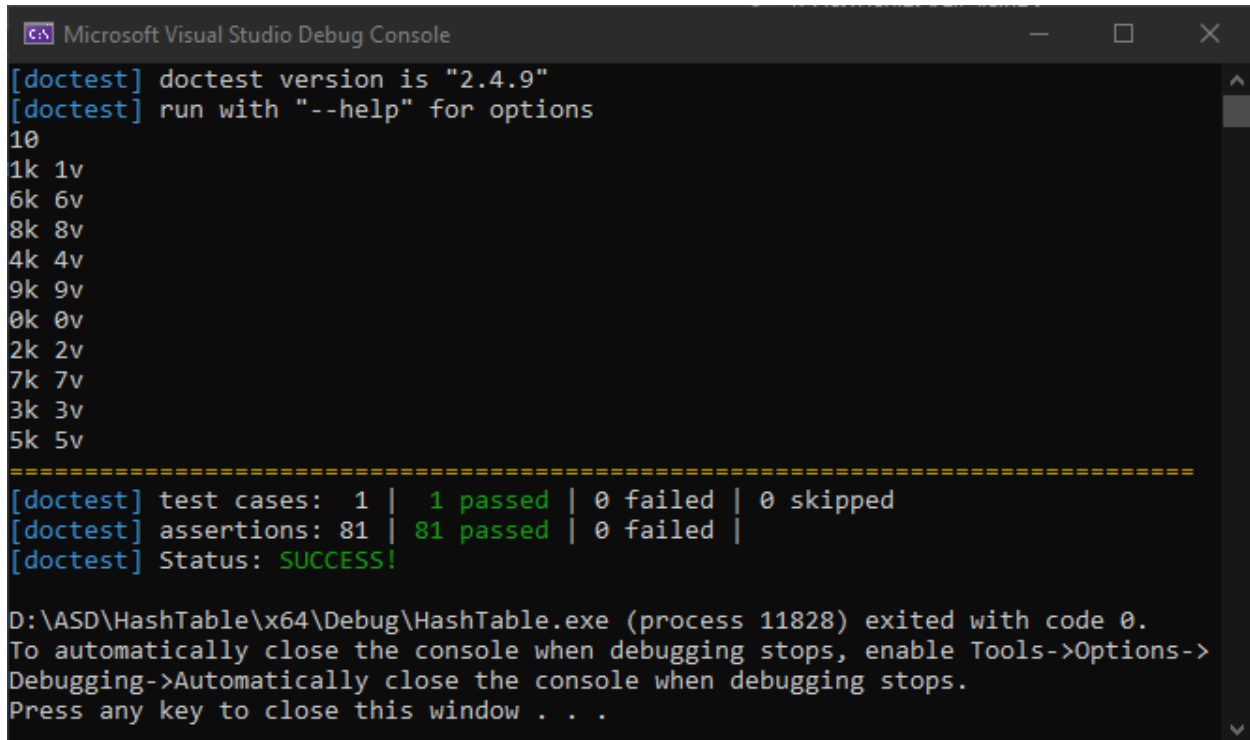
Складність роботи методів:

- Add : $O(1 + \log_2 n/s)$
- Find : $O(1 + \log_2 n/s)$
- Replace : $O(1 + \log_2 n/s)$
- Remove : $O(1 + \log_2 n/s)$
- Clear : $O(n)$
- Output : $O(n)$
- GetCount : $O(1)$

Просторова складність таблиці: $O(n)$

Приклад:

Щоб переконатись, що всі методи класу Хеш-Таблиця працюють правильно, в програмі написані юніт-тести. Усі вони проходять успішно:

A screenshot of the Microsoft Visual Studio Debug Console window. The title bar shows the Visual Studio icon and the text "Microsoft Visual Studio Debug Console". The console output is as follows:

```
[doctest] doctest version is "2.4.9"
[doctest] run with "--help" for options
10
1k 1v
6k 6v
8k 8v
4k 4v
9k 9v
0k 0v
2k 2v
7k 7v
3k 3v
5k 5v

=====
[doctest] test cases:  1 |  1 passed | 0 failed | 0 skipped
[doctest] assertions: 81 | 81 passed | 0 failed |
[doctest] Status: SUCCESS!

D:\ASD\HashTable\x64\Debug\HashTable.exe (process 11828) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->
Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Висновок:

В цілому, таблиці є корисною структурою для організації та доступу до даних, і їх можна реалізувати за допомогою різних підходів, включаючи однозв'язні списки, двозв'язні списки, масиви та дерева.