Datenstrukturen und Algorithmen, Serie 6

Nûr Johanna Studer 12-122-438, Dario Marti 18-109-967

1.)

a.) Keys mit zugehörigen Werten:

2022 => 340

2023 => 144

2024 => 461

2025 => 265

b.) Key welcher eine Postleitzahl von Bern ist und eine Kollision verursacht:

Gefunden mit ausprobieren.

2021 => 23

3008 => 23

2.)

Die Zeitkomplexität verbessert sich bei der Suche zu log(n), da die Hashs mit Binary-Search durchsucht werden können. *INSERT* verändert sich in der Komplexität nicht. *DELETE* ist konstant O(1) und nicht mehr nur im Mittel.

3.)

a.) Lineares Sondieren:

| 56 | 13 | 24 |  | 33 |  | 18 | 8 | 23 | 7 | 44 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

b.) Quadratisches Sondieren mit und

| 44 | 56 | 24 | 13 | 33 |  | 23 | 18 | 7 | 19 |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

c.) Doppeltes Hashing mit

| 44 | 56 | 24 | 33 |  |  | 23 | 18 | 19 | 13 | 7 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

4.) Sei eine Hash-Tabelle und eine Freiliste.

steht für einen freien Eintrag und für einen besetzten.

Damit ein Eintrag in die Freiliste eingetragen werden kann, wird im Eintrag auf gesetzt und dann mit in eingetragen.

Sei nun ein Element. Wenn es für keinen Eintrag in hat, so ist in enthalten. Falls ein existiert für das gilt, dass , dann gibt es einen Eintrag.

Es kann aber auch sein, dass also die Stelle besetzt ist mit einem anderen Element.

Für die Operationen gilt:

*SEARCH:* Überprüfe, ob für einer der oben genannten Fälle zutrifft. Wenn das Element enthalten ist, dann ist das Vorgehen gleich wie bei einer normalen Hash-Tabelle. Bei den anderen Fällen muss die Suche nicht fortgesetzt werden.

*INSERT:* Hier kann man 3 Fälle unterscheiden:

1. ist frei. Dann wird der Eintrag aus entfernt und dann in eingesetzt. Es gilt dann .
2. mit . Nehme das erste Element aus mit Index . Setze dann und .
3. mit . Setze für ein neues Element mit Index . Nehme nun das vorherige Element und setze dessen Wert auf . ist jetzt leer und Schritt 1.) kann wiederholt werden.

*DELETE:* Wenn kommt einfach in . Der Eintrag des

Vorgängers wird dann auf gesetzt. Sonst sei . und kommt in

Man braucht eine doppelt verlinkte Liste, da sonst ein Element immer mit dem Vorherigen verkettet ist. So müsste immer zuerst das vorherige Element gefunden werden, bevor es gelöscht werden könnte.