

MA 251 Data Structures
Laboratory Assignment 8
29-10-2019

Note: Upload your programs to the server (deadline: 4:30 pm)

1. Hashing with Chaining:

In this lab assignment, our goal is to implement a hash table with lists chaining. The hash table will store strings and it should support *add*, *delete* and *search* operations. You are already given the number of buckets m and the hash function h . It is a polynomial hash function

$$h(S) = \left(\sum_{i=0}^{|S|-1} S[i]x^{i \bmod p} \right) \bmod m$$

where $S[i]$ is the ASCII code of the i -th symbol of string S , $p = 1\,000\,000\,007$ and $x = 263$. Choose the value of $m = 10$. The operations supported are

- add <string>: insert string into the table. Ignore if already present.
- del <string>: Remove string from the table. Ignore the query if string not present
- find <string>: Output yes or no

Sample Output:

```
> add hello
> add world
> find World
no
> find world
Yes
```

2. Open addressing

In this lab assignment, you will implement a phone book manager to store <phone#, name> pairs. The pairs will be stored in a hash table using open addressing with quadratic probing. The phone# is used as the key. The hash function is given as

$$h(k, i) = (h_1(k) + i^2) \bmod m \text{ where } h_1 = ((ax + b) \bmod p) \bmod m$$

The phone# stored will not be more than 100, you can choose the value of m appropriately. Choose p a prime number between m and $2m$, $0 < a < p$ and $0 \leq b < p$.

The phone manager should support the following operations:

- add number name: If there exists user with same phone#, overwrite corresponding name
- del number: If no such number exists, ignore the query.
- find number: Should print the appropriate or *not found*.

Constraints: All phone numbers consist of decimal digits, they don't have leading zeros, and each of them has no more than 7 digits.

Sample Output:

```
> add 100 Police
> add 104 Ambulance
> add 9864021 Alice
> find 100
Police
> find 103
Not found
```