Data Structures and Algorithms

Mohammad GANJTABESH

Department of Computer Science,
School of Mathematics, Statistics and Computer Science,
University of Tehran,
Tehran, Iran.

mgtabesh@ut.ac.ir

Dept. of Computer Science
University of Tehran

Hashing

درهم سازی

Data Structures and Algorithms

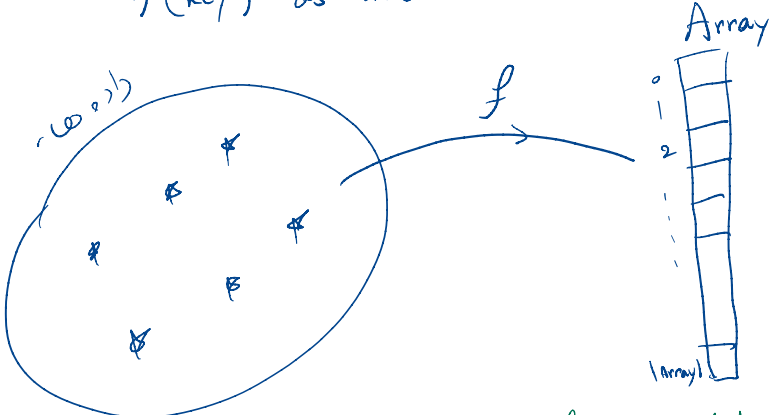
Undergraduate course

Mohammad GANJTABESH
mgtabesh@ut.ac.ir

Dept. of Computer Science
University of Tehran

Motivations

Hashing : storing data as <key, value> based on $f(\text{key})$ as index.



$\Rightarrow$ Efficient add, remove, find, update .... methods.

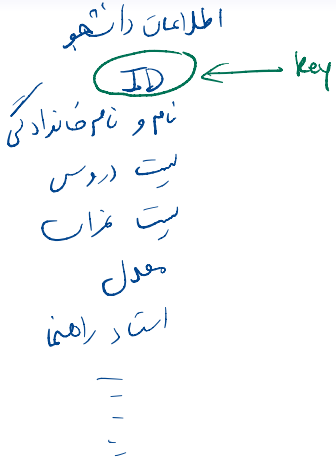
Hash function and its properties

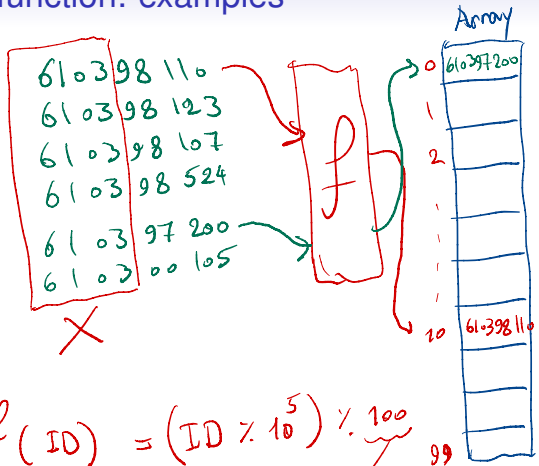
Hash function : key set ⟶ index set

Properties:

- depends on data properties.
- should be fast (efficient)
- equal data ⇝ same index.
- always returns the same index in different runs.
- minimizes the collision (uniformly distributes the data on Index set)
  - ↳ optimize the table size (prime numbers)

Hash function: examples





Folding

Hash function: examples

$$f(n) = \left[\sum \text{folds}(n)\right] \% |\text{table}|$$

0912 114 2224

09127563141

⋮

0935 514 0460

⋮

Hash function: examples

Strings :

"abcaaccad"

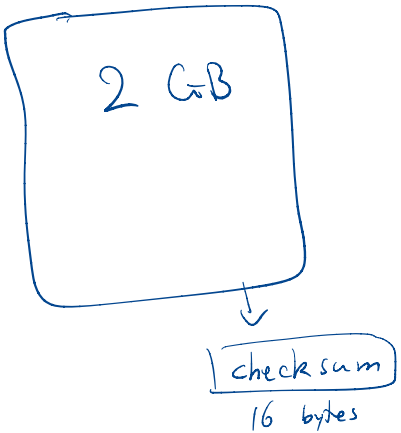
- sum of the ascii codes.

"this" , "hits"

collide

- sum of the ascii codes including positions.

"this" $\rightsquigarrow$ (t) $\times 1$ + (h) $\times 2$ + (i) $\times 3$ + (s) $\times 4$ $\neq$ (h) $\times 1$ + (i) $\times 2$ + (t) $\times 3$ + (s) $\times 4$



Resolve collisions

Load Factor : $\lambda = \frac{\text{number of items}}{\text{total size of the table}}$

$0 \leq \lambda \leq 1$

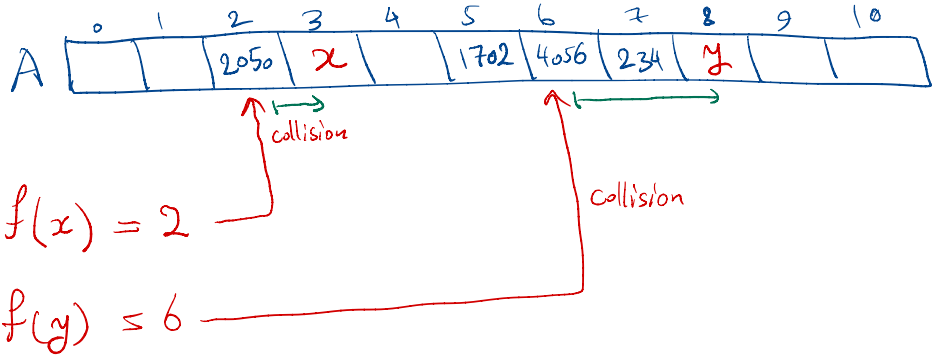
if $\lambda > \theta \Longrightarrow$ Rehashing. (using a longer table)

Resolve Collisions:

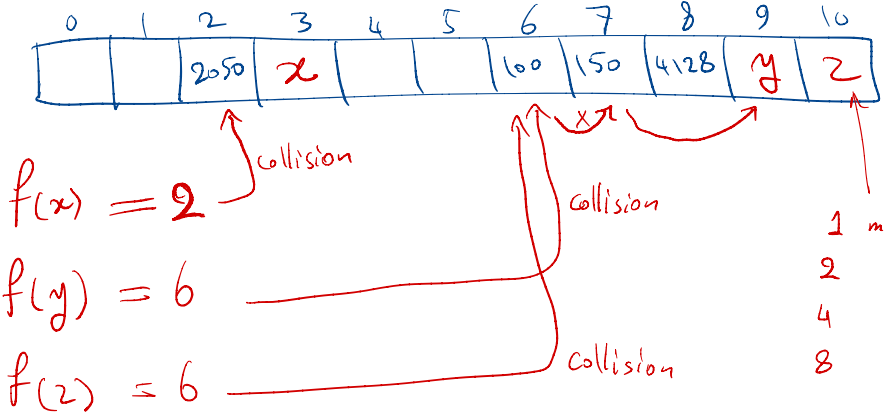
- Open addressing
  - linear probing
  - Quadratic "
  - Double hashing
- Chaining

Resolve collisions: Linear probing

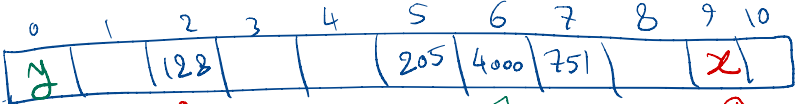
Using the first available position



Resolve collisions: Quadratic probing



Resolve collisions: Double hashing



collision

$f(x) = 2$

$g(x) = 7$

$f(x) + g(x) = 2 + 7 = 9$

> 0

collision

$f(y) = 6$

$g(y) = 5$

$\Longrightarrow 6 + 5 = 11$

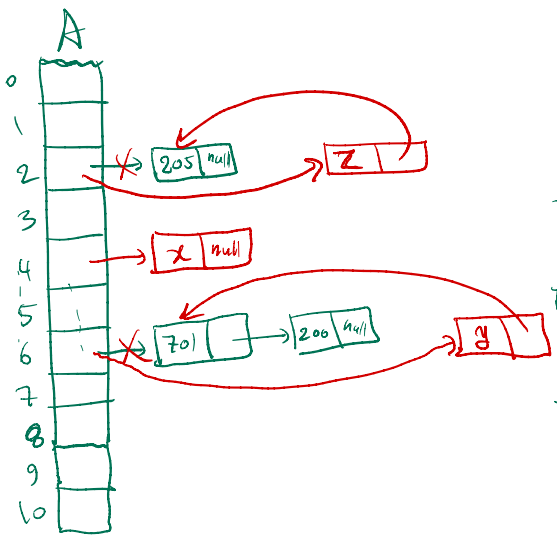
* using two different hash functions and the second one should be positive.

if collision : hash1(x) + hash2(x)

Resolve collisions: Chaining

$\lambda > 1$ ممکن

↳ using Linked List to store the data having the same index.



$f(x) = 4$

$f(y) = 6$

$f(z) = 2$