UNIVERSITY of WASHINGTON

CSS 343: Data Structures, Algorithms, and Discrete Mathematics II

Hashing

Version 1

Wooyoung Kim

Hash Table

- Average complexity
  - Insert/delete/retrieval of binary search trees?
  - Hash table: O(1)
- Worst-case complexity
  - Insert/delete/retrieval of binary search trees?
  - Hash table: O(n)
- Basic idea:
  - use a **hash function** to map the items into positions in an array
  - Problem: Collision, and clustering

Collisions

- **Collision**
  - When two keys map to the same location in the hash table
- **Flavors of Collision Resolution**
  - Closed hashing (Open Addressing)
    - Linear probing
    - Quadratic Probing
    - Double Hashing
  - Open hashing (Reconstructing the hash table)
    - Buckets
    - Separate chaining

Closed hashing

- When a collision occurs, find another bucket for the new object
  - Probing distance function D(i) to resolve collisions
    - i: number of collisions in the current attempt
  - Next bucket to exam: (h(x) + D(i)) mod B
    - B: total # of buckets in a hash table
    - Generally suggested B: prime number and at least 2*expected # of items
- D(i)
  - Linear probing: D(i) = i
  - Quadratic probing: D(i) = i^2
  - Double hashing: D(i) = i*h_2(x)

Linear probing

- After i collision, (h(x) + D(i)) mod B
  - (h(x) + D(i)) mod B, where D(i) = i
  - That is, if A[i] is occupied
    - Try A[(i+1) mod N], if still occupied
    - Try A[(i+2) mod N],

    And so on,
- Assume h(x) = x%10, N = 10 (hash table size)
- Example
  - 37, 98, 107, 20, 57
- Practice
- Suffer from Primary Clustering Problem
  - "Long runs of occupied slots built up (near the h(x) positions), increasing the average search time."

0	20
1	57
2	
3	
4	
5	
6	
7	37
8	98
9	107

Quadratic probing

- After i collision, (h(x) + D(i)) mod B
- (h(x) + D(i)) mod B, where D(i) = i^2
  - That is, if A[i] is occupied
    - Try A[i+1^2 mod N], if still occupied
    - Try A[i+2^2 mod N],

    And so on,
- Assume h(x) = x%10, N = 10 (hash table size)
- Example: 37, 18, 107, 57
- Although fewer, suffer from Secondary Clustering
  - elements that hash to the same position will probe the same alternate cells (same sequence is followed to handle collision)
  - The size of the hash table should be a prime number to avoid repeats

0	
1	107
2	
3	
4	
5	
6	57
7	37
8	18
9	

Double hashing

- (h(x) + D(i)) mod B, where D(i) = i*h_2(x)
- Typically, h_2(x) = r – (x % r); r is a prime number smaller than B

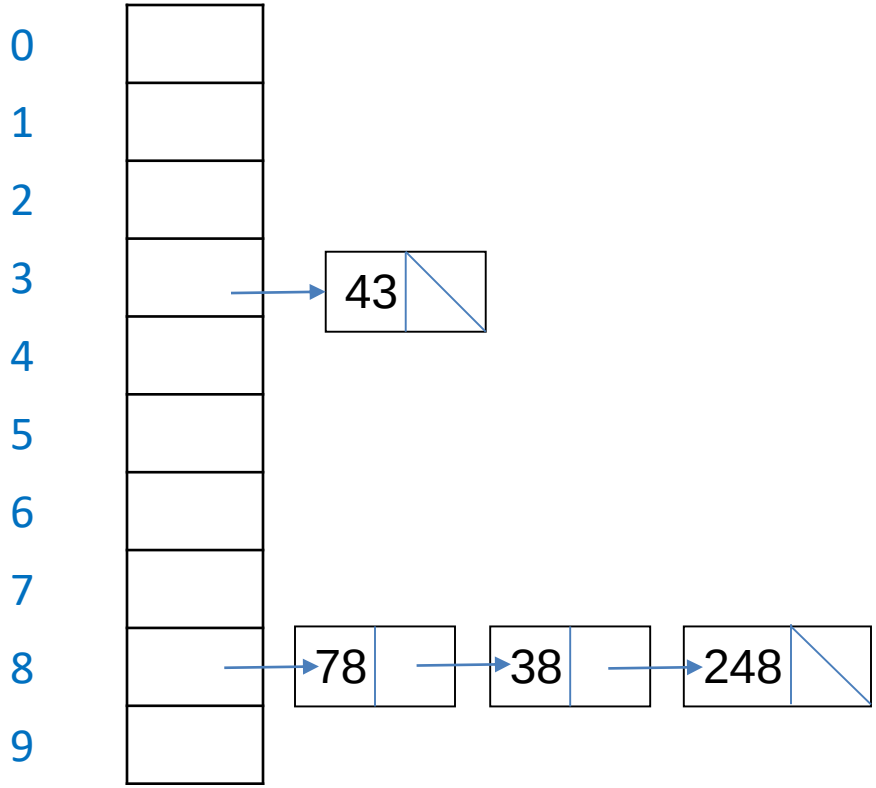
- Example
  - Assume h(x) = x % 10; h_2(x) = 7 – (x % 7), hash 89, 18, 49, 58, 50, 60, 23
- **If B is non-prime number,** possibly lead to **cycle of repeated visits**

Fail on 23

0	50
1	
2	60
3	58
4	
5	
6	49
7	
8	18
9	89

Open hashing

- each bucket of the hash table stores a linked list of the objects that hash to that bucket
- A simple example
  - h(x) = x % 10, hash 78, 38, 43, 248
- Practice



Define hash function

Define your own hash function:

How to write a hash function to quickly find the following educational programs at UW Bothell in a hash table (BBUS, BEDUC, BES, BHLTH, BLS, BNURS, BPOLST, CSS)

Note: A good hash function is

1) Easy to compute

2) Will evenly distribute the possible data

Try:

1) Use a prime number as a size of hash table

2) Avoid collision

Define hash function

Define your own hash function:

How to write a hash function to quickly find the following educational programs at UW Bothell in a hash table (BBUS, BEDUC, BES, BHLTH, BLS, BNURS, BPOLST, CSS)

WAY #1:

sum up ascii values of each character and use table size 17

A -> 65

B -> 66

…

Z -> 90

CSS: C(67)+S(83)+S(83) => 233 %17 = 12

After finish it, you can shift the values so that the smallest number be zero

Define hash function

Define your own hash function:

How to write a hash function to quickly find the following educational programs at UW Bothell in a hash table (BBUS, BEDUC, BES, BHLTH, BLS, BNURS, BPOLST, CSS)

WAY #2:

2-D array with its length as the row

- Total of 5 rows (index starts from 1)
- The second letter, asci value % 5 (5 = column)

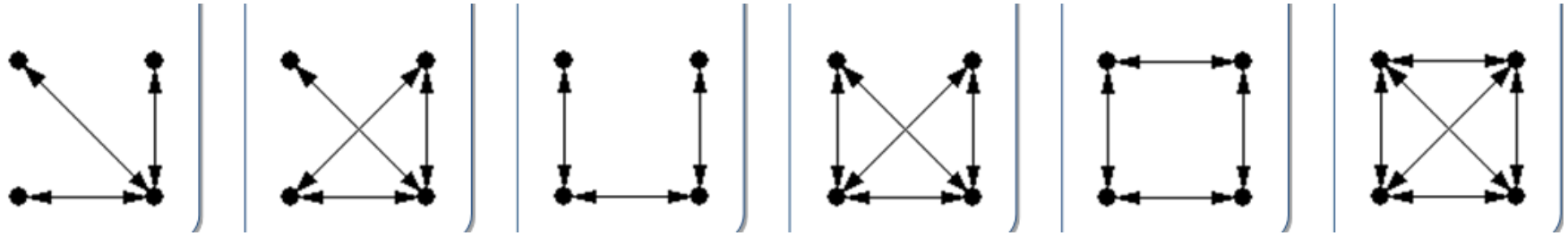
CSS

row=3, column = 83%5 = 3.

So, CSS is in array[3][3]

Graph hash function

How to define a hash function for a graph?



Design your own hash function to store all the 6 types of graphs.

Reference: g6 format see Files/additional materials/format.txt on Canvas