CS010C

No.10 Lab

Hash Lab Demo

- Attendance, code log on AWS, submission on CANVAS
- Always remember "%"
- 4 Hash Functions (separate chaining)
  - $h(k) = k[0]$ (ASCII 1st letter of key)
  - $h(k) = k[0] + 27 * k[1] + 729 * k[2]$
  - $h(k) = \sum_i k[i] * 37^i$
  - hash function chosen by student (student explanation)
- Open Hashing Collision Testing (with probing)
  - probing sequence: i, i^2, i^3
  - collisions versus load factor (70%, 80%, 90%)

UCR_CS010C_25U > Lab10 > sample_hash.out

```
0:  nnnnn,
1:  ooooo,
2:  ppppp,
3:  qqqqq,
4:  hello, happy, heath, harps, rrrrr,
5:  iiiii, sssss,
6:  jjjjj, ttttt,
7:  kkkkk, uuuuu,
8:  lllll,
9:  mmmmm,
```

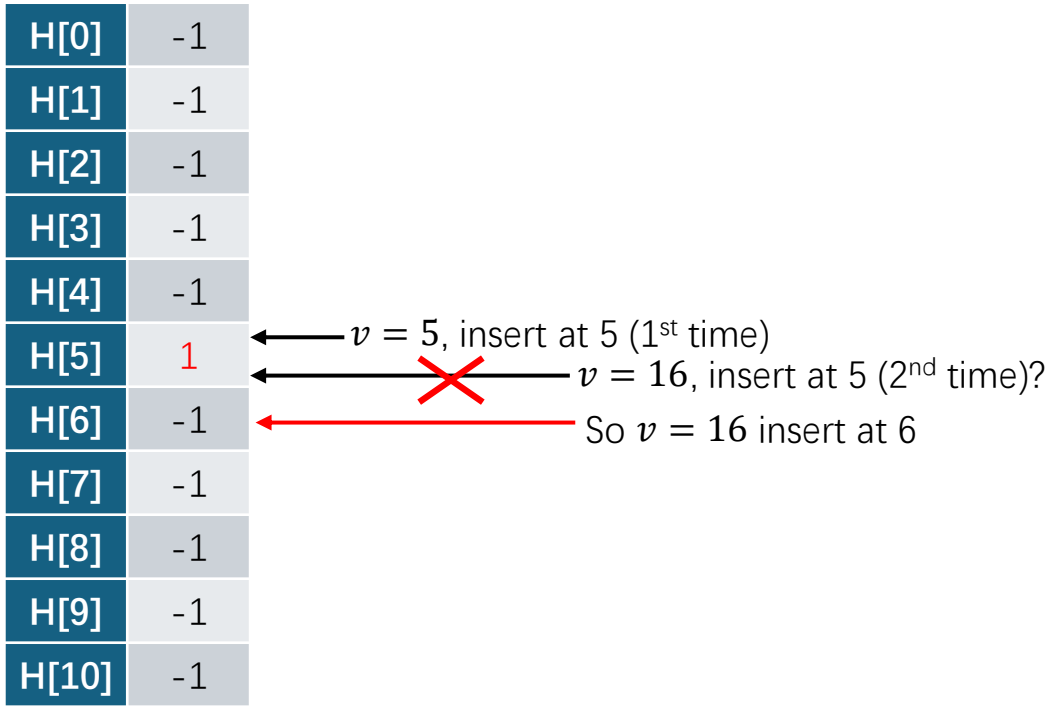
```
❯ g++ collisions.cpp
❯ ./a.out 10007 7000
Here is the number of collisions for each type of probing.
linear = 6014
quadratic = 5587
cubic = 5530
❯ ./a.out 10007 8000
Here is the number of collisions for each type of probing.
linear = 9970
quadratic = 8877
cubic = 8825
❯ ./a.out 10007 9000
Here is the number of collisions for each type of probing.
linear = 18098
quadratic = 15487
cubic = 15160
~/UCR_CS010C_25U/Lab9  main
```

Review: Probing

- Initialize: all -1s(no item in table)
- Insert random number
  - Check the correspond value is not 1
  - Set corresponding value to 1
- Collision: $h[pos] == 1$
  - already occupied, can't insert here
  - probe the next slot ($pos + offset$)
  - If another collision occurs, continue probing forward
    - Linear: $offset = i,$ $i = 1,2,3 \ldots$
    - Quadratic: $offset = i^2$ OR $c_1 i + c_2 i^2,$ $i = 1,2,3, \ldots$
    - Cubic: $offset = i^3$ OR $c_1 i + c_2 i^2 + c_3 i^3,$ $i = 1,2,3, \ldots$

simplest case, OR you can choose different coefficients c_i

Linear hash table



Thanks for a great semester, everyone.
Best of luck on your finals!