Due Date: 04-26-2019 (F), BEFORE the class begins

This assignment covers textbook Chapter 11 and Chapter1~8.

1. **Hash Table** (20 points)
Exercises 11.2-1, page 261

2. **Hash Function** (60 points, 15 for each sub-question)
Consider inserting keys 3,4,2,5,1 in the order given into a hash table of length $m = 5$ using hash function $h(k) = k^2 \bmod m$ (k^2 is the auxiliary function).

(1) Using $h(k)$ as the hash function, illustrate the result of inserting these keys using chaining. Also, compute the load factor α for the hash table resulting from the insertions.

(2) Using $h(k)$ as the primary hash function, illustrate the result of inserting these keys using open addressing with linear probing.

(3) Using $h(k)$ as the primary hash function, illustrate the result of inserting these keys using open addressing with quadratic probing, where $c1=1$ and $c2=2$.

(4) What different values can the hash function $h(k) = k^2 \bmod m$ produce when $m = 11$? Carefully justify your answer in detail.

3. **Algorithm Design** (20 points)
Consider an unsorted array A that contains n *(n>=2)* distinct natural numbers, design an *efficient* algorithm to determine if the array contains two integers such that they add up to a specific target number s. That is: if we can find A[i] + A[j] == s ($1 \le i, j \le n, i \ne j$), s is an integer, the algorithm should return TRUE, otherwise return FALSE.

Design requirement: the *efficient* algorithm you are going to design should provide a linear running time, rather than a $O(n^2)$ running-time brute-force solution or a O(nlgn) solution. You may use the algorithms that we learned in the textbook.

(1) Write the Pseudo-code *(please use textbook conventions)* (15 points)
(2) Justify the running time of the algorithm (5 points).

Algorithms -- COMP.4040 Honor Statement
(Courtesy of Prof. Tom Costello and Karen Daniels with modifications)

Must be attached to each submission

Academic achievement is ordinarily evaluated on the basis of work that a student produces independently. Infringement of this Code of Honor entails penalties ranging from reprimand to suspension, dismissal or expulsion from the University.

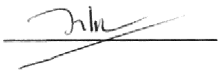
Your name on any exercise is regarded as assurance and certification that what you are submitting for that exercise is the result of your own thoughts and study. Where collaboration is authorized, you should state very clearly which parts of any assignment were performed with collaboration and name your collaborators.

In writing examinations and quizzes, you are expected and required to respond entirely on the basis of your own memory and capacity, without any assistance whatsoever except such as what is specifically authorized by the instructor.

I certify that the work submitted with this assignment is mine and was generated in a manner consistent with this document, the course academic policy on the course website on Blackboard, and the UMass Lowell academic code.

Date: 04/26/2019

Name (please print): DANGNHI NGO

Signature: 

88/100

1/ Hash Table

Exercise 11.2 - 1

Use hash function h to hash

distinct key n

array T

$T.length = m$

With simple uniform hashing, expected number of collisions:

$\Rightarrow$ expected cardinality of $\{\{k, l\} : k \neq l \,/\, h(k) = h(l)\}$

Solution:

Load factor: $\alpha = n/m$ (expected number of collisions)

Therefore, load factor means how many keys hashes to (1 - sol) of the table

With simple uniform hashing,

$$\alpha = P_r\{h(k) = h(l)\} = 1/m \text{ and}$$

$$\therefore E[\alpha] = 1/m$$

Total numbers of collisions A,

The expected number of collisions:

$$E[A] = E\left[\sum_{k \neq l} \alpha\right]$$

$$= \sum_{k \neq l} E[\alpha] \quad \text{- linearity of expectation}$$

$$= \sum_{k \neq l} 1/m$$

$$= \binom{n}{2} \frac{1}{m}$$

$$= \frac{n!}{2!(n-2)!} \cdot \frac{1}{m}$$

$$= \frac{n \times (n-1) \times (n-2) \times \dots \times 1}{2 \times (n-2) \times (n-3) \times \dots \times 1} \times \frac{1}{m}$$

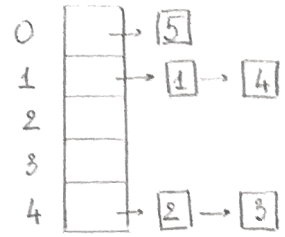
$$= \frac{n(n-1)}{2} \times \frac{1}{m}$$

$$= \frac{n(n-1)}{2m}$$

2/ Hash Function

58 Consider inserting keys 3, 4, 2, 5, 1 in the order given into a hash table of length $m = 5$ using hash function $h(k) = k^2 \bmod m$

(1) Using chaining



$\alpha = n/m = 5/5 = 1$

$h(k) = k^2 \bmod m$

$h(1) = 1^2 \bmod 5 = 1$

$h(2) = 2^2 \bmod 5 = 4$

$h(3) = 3^2 \bmod 5 = 4$

$h(4) = 4^2 \bmod 5 = 1$

$h(5) = 5^2 \bmod 5 = 0$

(2) Using open addressing with linear probing

$h(k, i) = (h(k) + i) \bmod m$

$= (k^2 + i) \bmod 5$ (because $m = 5$)

Key	Function	Value
3	$h(3,0) = 3^2 \bmod 5$ $= 9 \bmod 5$	$\Rightarrow$ 4
4	$h(4,0) = 4^2 \bmod 5$ $= 16 \bmod 5$	$\Rightarrow$ 1
2	$h(2,0) = 2^2 \bmod 5$ $= 4 \bmod 5 = 4$ $h(2,1) = (2^2 + 1) \bmod 5$ $= 0$	$\Rightarrow$ 0
5	$h(5,0) = 5^2 \bmod 5 = 0$ $h(5,1) = (5^2 + 1) \bmod 5 = 1$ $h(5,2) = (5^2 + 2) \bmod 5 = 2$	$\Rightarrow$ 2
1	$h(1,0) = 1^2 \bmod 5 = 1$ $h(1,1) = (1^2 + 1) \bmod 5 = 2$ $h(1,2) = (1^2 + 2) \bmod 5 = 3$	$\Rightarrow$ 3

0	2
1	4
2	5
3	1
4	3

(3) Using open addressing with quadratic probing

where $c1 = 1$ and $c2 = 2$

$h(k,i) = (h(k) + c1\,i + c2\,i^2) \bmod m$

$= (k^2 + i + 2i^2) \bmod 5$ (because $m = 5$)

Key	Function	Value
3	$h(3,0) = (3^2 + 0 + 0) \bmod 5 = 4$	$\Rightarrow 4$
4	$h(4,0) = 4^2 \bmod 5 = 1$	$\Rightarrow 1$
2	$h(2,0) = 2^2 \bmod 5 = 4$ $h(2,1) = (2^2 + 1 + 2.1^2) \bmod 5 = 2$	$\Rightarrow 2$
5	$h(5,0) = 5^2 \bmod 5 = 0$	$\Rightarrow 0$
1	$h(1,0) = 1^2 \bmod 5 = 1$ $h(1,1) = (1^2 + 1 + 2\,1^2) \bmod 5 = 4 \bmod 5 = 4$ $h(1,2) = (1^2 + 2 + 2\,2^2) \bmod 5 = 11 \bmod 5 = 1$ $h(1,3) = (1^2 + 3 + 2\,3^2) \bmod 5 = 22 \bmod 5 = 2$ $h(1,4) = (1^2 + 4 + 2.4^2) \bmod 5 = 37 \bmod 5 = 2$ $h(1,5) = (1^2 + 5 + 2.5^2) \bmod 5 = 51 \bmod 5 = 1$ $h(1,6) = (1^2 + 6 + 2.6^2) \bmod 5 = 79 \bmod 5 = 4$	

0	5
1	4
2	2
3	
4	3

Therefore, $h(1,i) = (1 + i + 2.i^2) \bmod 5$

$h(1,7) = (1 + 7 + 2.7^2) \bmod 5 = 106 \bmod 5 = 1$

$h(1,8) = (1 + 8 + 2.8^2) \bmod 5 = 137 \bmod 5 = 2$

$h(1,9) = (1 + 9 + 2.9^2) \bmod 5 = 172 \bmod 5 = 2$

$h(1,10) = (1 + 10 + 2.10^2) \bmod 5 = 211 \bmod 5 = 1$

$h(1,11) = (1 + 11 + 2.11^2) \bmod 5 = 254 \bmod 5 = 4$

$h(1,12) = (1 + 12 + 2.12^2) \bmod 5 = 301 \bmod 5 = 1$

$\Rightarrow h(1,i)$ is impossible to get slot value 3, thus C1 and C2 values need to be changed

(4) When $m = 11$, what different values the hash function $h(k) = k^2 \bmod m$ can produce:

$h(k) = k^2 \bmod 11$

Key	Function	Value
3	$h(3) = 3^2 \bmod 11 = 9 \bmod 11$	9
4	$h(4) = 4^2 \bmod 11 = 16 \bmod 11$	5
2	$h(2) = 2^2 \bmod 11 = 4 \bmod 11$	4
5	$h(5) = 5^2 \bmod 11 = 25 \bmod 11$	3
1	$h(1) = 1^2 \bmod 11 = 1 \bmod 11$	1

-2

0

With this $h(k)$ function, we need total 11 slots for 5 elements: $\alpha = 5/11$, which means that there would be at least 6 empty slots all the time and that's wasting of memory

Therefore, all elements should have their own slots, and it will be easier to search or delete elements from hash table

0	/
1	1
2	2
3	5
4	/
5	4
6	/
7	/
8	/
9	3
10	/

3/ Algorithm Design

a/ Algorithm:

10

+ Sort the array in non-decreasing order
+ Initialize two index variables to find the candidate elements in the sorted array
+ Loop while and compare sum of two natural numbers with a specific target number s

```
Find Sum (A, n, s)
    // sort first
    MERGE_SORT (A, 1, n)                    ]  θ(nlgn)
    or HEAP_SORT (A)
    // initialize two index variables
    i = 1 , J = n                           ]
    // find a pair                          ]
    while ( i < J )                         ]
        if ( A[i] + A[J] == s)              ]  θ(n)
            return TRUE                     ]
        else if (A[i] + A[J] > s)           ]
            i = J - 1                       ]
        else i ++                           ]
    return FALSE                                θ(1)
```

b/ Running time of the algorithm

$T(n) = \theta(n\lg n) + \theta(n) + \theta(1)$

$= \theta(n\lg n)$

design $O(n)$