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: 6/25/2019

Name (please print): PHONG VO

Signature: ______

Due Date: 06-26-2019 (W), 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 ($1 \le i, j \le n, i \neq j$), 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 $O(n\lg n)$ 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).

1. Using probability

Hash table has length $m \Rightarrow$ Sample space $= m$

Each output of a hash function has 1 result to load in the hash table. $\Rightarrow$ event $X = 1$

probability of P(keys hashed to same value) $= \frac{1}{m}$

Lemma: $= X_{ij}$

X: numbers of collisions

$$X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}$$

$$E[X] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}]$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{m}$$

$$= \frac{1}{m} \sum_{i=1}^{n-1} n-(i+1)+1 = \frac{1}{m} \sum_{i=1}^{n-1} (n-i)$$

$$= \frac{1}{m} \left[(n-1) \cdot \frac{n-(n-1)+n-1}{2} \right]$$

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

HW 8

(2) $h(k) = k^2 \bmod (m=5)$ $\langle 3, 4, 2, 5, 1 \rangle$

1/ Using chaining:

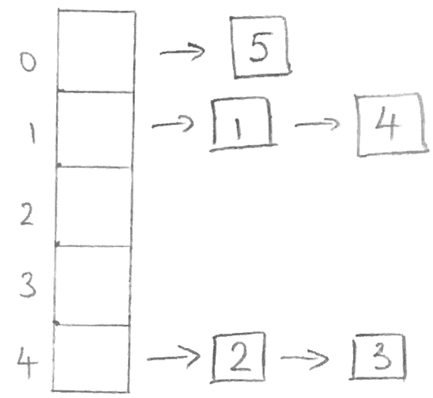
$h(3) = 3^2 \bmod 5 = 4$

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

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

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

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



$\alpha = \frac{n}{m} = \frac{5}{5} = 1$

2/ Using open addressing with linear probing:

Key = 3: $h(3,0) = 3^2 + 0 \bmod 5 = 4$

Key = 4: $h(4,0) = 4^2 + 0 \bmod 5 = 1$

Key = 2: $h(2,0) = 2^2 + 0 \bmod 5 = 4$ → collision

$h(2,1) = 2^2 + 1 \bmod 5 = 0$

Key = 5: $h(5,0) = 5^2 + 0 \bmod 5 = 0$ → collision

$h(5,1) = 5^2 + 1 \bmod 5 = 1$ → collision

$h(5,2) = 5^2 + 2 \bmod 5 = 2$

Key = 1: $h(1,0) = 1^2 + 0 \bmod 5 = 1$ → collision

$h(1,1) = 1^2 + 1 \bmod 5 = 2$ → collision

$h(1,2) = 1^2 + 2 \bmod 5 = 3$

0	2
1	4
2	5
3	1
4	3

HW 8

3/ Using open addressing with quadratic probing $c_1 = 1$ and $c_2 = 2$

$$h(k,i) = (h(k) + c_1 i + c_2 i^2) \bmod m$$

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

Key = 3: $h(3,0) = (3^2 + 0 + 0^2) \bmod 5 = 4$

Key 4: $h(4,0) = (4^2 + 0 + 0^2) \bmod 5 = 1$

Key 2: $h(2,0) = (2^2 + 0 + 0^2) \bmod 5 = 4$ collision

$h(2,1) = (2^2 + 1 + 2.1^2) \bmod 5 = 2$

Key 5: $h(5,0) = (5^2 + 0 + 0^2) \bmod 5 = 0$

Key 1: $h(1,0) = (1^2 + 0 + 0^2) \bmod 5 = 1 \to$ collision

$h(1,1) = (1^2 + 1 + 2.1^2) \bmod 5 = 4 \to$ collision

$h(1,2) = (1^2 + 2 + 2.2^2) \bmod 5 = 1 \to$ collision

$h(1,3) = (1^2 + 3 + 2.3^2) \bmod 5 = 2 \to$ collision

$h(1,4) = (1^2 + 4 + 2.4^2) \bmod 5 = 2 \to$ collision

$h(1,5) = (1^2 + 5 + 2.5^2) \bmod 5 = 1 \to$ collision

$h(1,6) = (1^2 + 6 + 2.6^2) \bmod 5 = 4 \to$ collision

Keep increasing i value but there will be no slot 3. Thus, c_1 and c_2 need to be changed.

4/ $h(k) = k^2$ whereas k can be expanded: $k = 11a + b$ ($a \geq 0$, $b \in [0..10]$)

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

$\Leftrightarrow (11a+b)^2 \bmod 11 \iff (11^2a^2 + 2 \times 11 \times ab + b^2) \bmod 11$

$\Leftrightarrow b^2 \bmod 11 \iff [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]$ (mod 11)

Thus, possible outcomes = $\langle 0, 1, 3, 4, 5, 9 \rangle$

HW 8

(3) Example $A = \langle 1, 2, 3, 4, 5 \rangle$ $S = 6$

1/ $\Rightarrow$ Returns TRUE because $A[2] + A[4] = 6$

Sum_seeking(A, n, s) // $n = A.size$, S : targeted sum

(1) Create a hash table T // $T.size = s+1$, hash function $h(k) = s - k$

(2) Seek = FALSE

$O(n+1)$
(3) for $i = 1$ to n
(4) if $A[i] \leq s$
(5) CHAINED-HASH-INSERT$(T, A[i])$ // insert every // element of A into the hash T

$O(n+1)$
(6) for $i = 1$ to n
(7) if $(A[i] \leq s)$ and (CHAINED-HASH-SEARCH$(T, s - A[i])$)
(8) Seek = TRUE

(9) return seek

2/ The running time of the algorith is $O(n+1) + O(n+1) = O(n)$

100/100