Robert Frenken

Homework 6

1. chained hashing - array of lists

hash function $h(k) = 4 \cdot k \bmod 7$

Insert $(5, D_5)$ -> $20 \bmod 7 = 6$

Insert $(10, D_{10})$ -> $40 \bmod 7 = 5$

Insert $(12, D_{12})$ -> $48 \bmod 7 = 6$

Insert $(15, D_{15})$ -> $60 \bmod 7 = 4$

Insert $(4, D_4)$ -> $16 \bmod 7 = 2$

Insert $(40, D_{40})$ -> $200 \bmod 7 = 4$

Insert $(3, D_3)$ -> $12 \bmod 7 = 5$

		$(5, D_5)$		$(15, D_{15})$, $(40, D_{40})$	$(10, D_{10})$, $(3, D_3)$	$(5, D_5)$, $(12, D_{12})$
Index:	0	1	2	3	4	5	6

2.

Insert $(3, De)$ $\quad h(K_{3,j}^{(3,0)}) = (4 \cdot 3 + 5 \cdot 0) \bmod 7$

$= 5$ -> coll.

j++ $\quad h(3,1) = (4 \cdot 3 + 5 \cdot 1) \bmod 7 = 3$ -> coll.

j++ $\quad h(3,2) = (4 \cdot 3 + 5 \cdot 2) \bmod 7 = 1$ -> Insert

Insert $(6, Df)$ $\quad h(6,0) = (4 \cdot 6 + 5 \cdot 0) \bmod 7 = 3$ -> coll

j++ $\quad h(6,1) = (4 \cdot 6 + 5 \cdot 1) \bmod 7 = 1$ -> coll.

j++ $\quad h(6,2) = (4 \cdot 6 + 5 \cdot 2) \bmod 7 = 6$ -> coll.

j++ $\quad h(6,3) = (4 \cdot 6 + 5 \cdot 3) \bmod 7 = 4$ -> Insert

0	1	2	3	4	5	6
D_a	D_e		D_b	D_f	D_c	D_d

3. size = 77 $h(k) = 35 \cdot k \bmod 77$ Chained hashing

h is not a good hash function, as since it's not a prime number, there could be a pattern that would cause many collisions, increasing the runtime. A better hash function would have a large prime number as the size, and implement a multiplicative or universal hashing instead of division hashing.

4. Prob4(Array<Integer> A, Array<Integer> B, Integer n)

a)

```
Hashmap. Init()
for i <- 1 to n do
    num <- B[i]
    h(num) <- hash Function
    Hashmap. Insert( h(num), num)
count <- 0
for i <- 1 to n do
    numA <- A[i]
    h(numA) <- hashFunction
    if Hashmap. Member( h(numA) )
        Count ++

return count
```

b) $ET(n) = \sum_{i=1}^{n} [ET_{insert}(s) + ET_{member}(s)] = \sum_{i=1}^{n} c = cn \in \Theta(n)$

Chained Hashing: $T_{in}(s) = c$ $T_{mem}(s) = c's$ $ET_{in}(s) = c$ $ET_{mem}(s) = c'$

c)

$$T(n) = \sum_{i=1}^{n} [T_{in}(s) + T_{mem}(n)] = \sum_{i=1}^{n} [c + ci] = cn + c\frac{n(n+1)}{2}$$

$$\in \Theta(n^2)$$

5.

```
Prob 5 (Array A, Integer n)

Hash.Init()

for i <- 1 to n do

    h(A[i]) <- hashFunction
    occurance <- 1
    if (Hash.member(A[i])
        Hash.Retrieve(A[i], count)
        count ++
        Hash.Insert(A[i], count)
    else
        Hash.Insert(A[i], occurance)

for i <- 1 to Hash.size do
    for j <- 1 to Hash.list.size do
        print(Hash.list[j])
```

b) $T_{insert}(s) = cs$ $\quad ET_{ins}(s) = c$

$T_{mem}(s) = c's^2$ $\quad ET_{mem}(s) = c'$

$$\sum_{i=1}^{n} [ET_{ins}(s) + ET_{mem}(s)] = cn \in \Theta(n)$$

c) $$\sum_{i=1}^{n} c + \sum_{i=1}^{n} \sum_{j=1}^{n} c = cn + cn^2 \in \Theta(n^2)$$