CS-3305A
Assignment 3
Ian Borwick-250950449

1. Hash table of size N=7, h(k)=k mod 7, elements ={19,27,12,47,15} using separate chaining to handle collisions

0	
1	
2	
3	
4	
5	19
6	

Insertion of 19
H(19)=19%7=5

0	
1	
2	
3	
4	
5	19
6	27

Insertion of 27
H(27)=27%7=6

0		
1		
2		
3		
4		
5	19	12
6	27	

Insertion of 12
H(12)=12%7=5

0			
1			
2			
3			
4			
5	19	12	47
6	27		

Insertion of 47
H(47)=47%7=5

0			
1	15		
2			
3			
4			
5	19	12	47
6	27		

Insertion of 15
H(15) = 15%7=1

2. Hash table of size N=7, h(k)=k mod 7, elements ={19,27,12,47,15} using linear probing to handle collisions

0	
1	
2	
3	
4	
5	19
6	
Insertion of 19 H(19)=19%7=5	

0	
1	
2	
3	
4	
5	19
6	27
Insertion of 27 H(27)=27%7=6	

0	12
1	
2	
3	
4	
5	19
6	27
Insertion of 12 H(12)=12%7=5 Collison H(12)+ 1= 6 Collision H(12)+2=0	

0	12
1	47
2	
3	
4	
5	19
6	27
Insertion of 47 H(47)=47%7=5 Collison H(47)+ 1= 6 Collision H(47)+2=0 Collision H(47)+3=1	

0	12
1	47
2	15
3	
4	
5	19
6	27
Insertion of 15 H(15)=15%7=1 Collision H(15)+1=2	

3. Hash table of size N=7, h(k)=k mod 7, elements ={19,27,12,47,15} using double hashing to handle collisions

0	
1	
2	
3	
4	
5	19
6	
Insertion of 19 H(19)=19%7=5	

0	
1	
2	
3	
4	
5	19
6	27
Insertion of 27 H(27)=27%7=6	

0	
1	12
2	
3	
4	
5	19
6	27
Insertion of 12 H(12)=12%7=5 Collison H(12)=5-(12%5)=3 H(12)=1	

0	
1	12
2	
3	
4	47
5	19
6	27
Insertion of 47 H(47)=47%7=5 Collison H(47)=5-(47%5)=3 H(47)=1 Collision H(47)+3 =4	

0	
1	12
2	15
3	
4	47
5	19
6	27
Insertion of 15 H(15)=15%7=1 Collision H(15)=5-(15%5)=5 H(15)=6 Collision H(15)+5=4 Collision H(15)+5=2	

4. F(1)=3
   F(n)=F(n-1)+2n+1

   Build
   F(n)=f(n-1)+2n+1
   F(n-1)=((f(n-2)+2(n-1)+1)+2n+1 = f(n-2)+2n-1+2n+1 = f(n-2)+4n
   F(n-2)=(f(n-3)+2(n-2)+1)+4n = f(n-3)+2n-3 +4n = f(n-3)+6n-3
   F(n-3) = (f(n-4) +2(n-3) +1)+6n-3 = f(n-4)+2n-5+6n-3 = f(n-4)+8n-8
   F(n-4) = (f(n-5) +2(n-4)+1) +8n-8 = f(n-5)+2n-8+1+8n-8 = f(n-5)+10n-15

   Expand
   F(n-1) = f((n-1)-1)+2(n-1) +1
   F(n-2) = f((n-3)+2(n-2)+1
   F(n-3) = f(n-4)+2(n-3)+1
   F(n-4) = f(n-5)+2(n-4)+1

   F(n) = f(n-10+2n+1
   = f(n-2)+4n
   = f(n-3)+6n -3
   =f(n-4)+8n – 8
   =F(n-5)+10n-15
   = F(n-i)+i(2n)+(c)

   $Xn=an^2+bn+c$
   X1=a+b+c=1
   X2=4a+2b+c=0
   X3=9a+3b+c

   X2-X1=3a+b 0-1=3a+b
   X3-X2=5a+b 0-3=5a+b

   -3+1=5a+b-(3a+b)
   -2=2a
   A=-1
   -1=3(-1)+b
   B=2
   A+b+c=1
   -1+2+c=1
   C=0
   $Ti=i^2+2i$
   Therefore $f(n)=f(n-1)+i(2n)+(-1^2+2i)$
   Since f(1)=3
   n-i=1
   i=n-1

$f(1)+(n-1)(2n)+(-(n-1)^2+2(n-1))$
$= 3+2n^2-2n+(-(n^2-2n+1)+2n-2)$
$= 3+2n^2-2n-n^2+2n-1+2n-2$
$= 2n^2-n^2-2n+2n+2n+3-2-1$
$=n^2-2n$
Therefore, the time complexity is $O(n^2)$

5.

a. **Algorithm** isSymmetric(r)
Input: root r of the tree
Output: true if the tree is symmetric false if it is not

```
if r= null
    return true
else
    x is child of r if r has child
    for each child c of r do
        if c.value !=x.value
            return false
    sym <- true
    for each child c of r do
        sym <- sym and isSymmetric(c)
return sym
```

b. Worst case is when the parent of the leaf is not symmetric

Let x be the max number of children for a parent in the provided tree:

\# of iterations =degree(r)
1.
C1 operation in base case; C1+C2+C3degree(r) in recursive case

2.
One call is performed per node

3.
$\sum$(leaves) C1 +(C2+C3xDegree(r)
= C1x#internal +C2xinternal + C3 $\sum$ degree(r)
= #leaves(n) +#internal is O(n)
O(n)