***Problem1***

1. From the definition, *R1 ∩ R2* = {*(s1, s2) : (s1, s2) R1 and (s1, s2) R2* }*.*

* Reflective: Since *R1* and *R2* are both reflexive, So for each *s S, (s, s) R,* then *(s, s) R1 ∩ R2.* Therefore, *R1 ∩ R2* is reflexive.
* Symmetric: Because *R1* and *R2* are both symmetric, so if *(s1, s2) R1 ∩ R2*, we have *(s1, s2) R1* and (s1, s2) ϵ R2, then *(s2, s1) R1, (s2, s1) R2*. So *(s2, s1) R1 ∩ R2*. Therefore, *R1 ∩ R2* is symmetric.
* Transitive: Let *s1, s2, s3 S*. Since *R1* and *R2* are both transitive, if *(s1, s2) R1*, *(s2, s3) R1*, then *(s1, s3) R1*. Similarly, *(s1, s3) R2. So (s1, s3) R1 ∩ R2.*

Therefore, R1 ∩ R2 is transitive.

Therefore, R1 ∩ R2 is an equivalence relation.

1. [x]1 = [s: s S and sR1x]

[x]2 = [s: s S and sR2x]

[x] = [s: s S and s(R1 ∩ R2)x] = [s: s S, x [x]1∩[x]2]

1. *R1∪R2* is not an equivalence relation.

* Reflective: Since *R1* and *R2* are both reflexive, So for each *s S, (s, s) R,* then *(s, s) R1∪R2*. Therefore, *R*1∪*R*2 is reflexive.
* Symmetric: Let (s1, s2) *R*1∪*R*2, if (s1, s2) R1, Because R1 is Symmetric, so (s2, s1) R1, then (s2, s1) *R*1∪*R*2 . If (s1, s2) R2, Because R2 is Symmetric, so (s2, s1) R2, then (s2, s1) ϵ *R*1∪*R*2 . Therefore, *R*1∪*R*2 is symmetric.
* Transitive: Let *S = {1, 2, 3}, R1 = {(1, 1), (2, 2), (3, 3), (1, 2), (2, 1)}, R2 = {(1, 1), (2, 2), (3, 3), (2, 3), (3, 2)}. R1∪R2 = {(1, 1), (2, 2), (3, 3), (1, 2), (2, 1), (2, 3), (3, 2)}.* For element (1, 2) and (2, 3) in R1∪R2, if R1∪R2 is transitive, (1, 3) will in *R1∪R2*. But (1, 3) is not in *R1∪R2*. There *R1∪R2* is not transitive.

Therefore, *R1∪R2* is not an equivalence relation.

***Problem2***

1. *R1; R2* = {*(a, c) : there is a b with (a, b) ∈ R1 and (b, c) ∈ R2*}. If *R1* and *R2* are both reflexive, for *s S*, let *(s, s) R1*, if *R1; R2*, then *(s, s) S*. So that *(s, s) ϵ R1; R2*. Therefore, if *R1* and *R2* are reflexive then *R1; R2* is reflexive.
2. Use a counterexample to disapprove. *R1* and *R2* are both symmetric, so let S = {1, 2, 3}, *R1 = {(1, 1), (1, 2), (1, 3)}, R2 = {(2,1), (2, 2), (2, 3)}. So R1; R2 = {(1, 1), (1, 2), (1, 3)}.*  *R1; R2* is not symmetric.
3. Use a counterexample from (b) to disapprove. *R1; R2* is not transitive.

***Problem3***

1. Use induction

Base case j = i: Clearly *R i = R i* . So *R j = R i* holds.

Inductive case: Assume that *R j = R i*  holds. We will show that R j+1 = R i holds.

R j+1 = R j ∪ (R; R j)

= R i ∪ (R; R j)

= R i+1

= R i

Therefore, by induction, *R j = R i* holds for all j i.

1. Use induction

When j i, clearly R j R i

Base case j = i: *R j = R i,* So *R j = R i* holds.

Inductive case: Assume that *R j*  *R i*  holds. We will show that R j+1 R i holds.

From (a) we have that *R j = R i* holds for all j i. so R j+1 = R j *R i*

Therefore, by induction, R j R i for all j

1. Use contradiction

Assume that Rk2 ≠ Rk2+1 (|S| = k)

So that (a, b) Rk2+1, but (a, b) ∉ Rk2

Rk2+1 = Rk2 ∪(R; Rk2), so (a, b) Rk2 ∪(R; Rk2)

So that (a, ck) R and (ck ,b) Rk2 (c≠a).

Rk2 = Rk2-1∪(R; Rk2-1), so (ck, b) Rk2-1∪(R; Rk2-1)

Because (a, ck) R, if (ck, b) Rk2-1, then (a, b) (R; Rk2-1) Rk2 , it is contrary to the assumption.

So (ck, b) ∉ Rk2-1, (ck, b) (R; Rk2-1)

……

(c1, c0) R and (c0, b) Rk2-1 (c1≠c0)

Because a S, |S| = k, and count(c0, c1, c2, ……ck) = k+1, so there exists ci = a, i [0, k]

This is contrary to the assumption. So the assumption is not true.

Therefore, if |S| = k, then Rk2 = Rk2+1

1. Use induction

Base case n = 0: For all m N, R 0+m = R m

= I ; Rm (From assignment1 problem 8(b))

= R0 ; Rm

So P(0) : R0;Rm = R0+m holds

Inductive case: Assume for all m, P(n) holds, that is Rn ; Rm = Rn+m , we will show that

P(n+1) : Rn+1 ; Rm = Rm+n+1 holds, that is :

Rn+1 ; Rm = (Rn ∪(R ; Rn)) ; Rm

= (Rn ; Rm) ∪ ((R ; Rn) ; Rm) (From assignment1 problem8 (c))

= (Rn ; Rm) ∪ (R ; (Rn ; Rm)) (From assignment1 problem8 (a))

= Rn+m ∪ (R ; Rn+m)

= Rn+m+1

= R(n+1)+m

So P(n+1) holds.

Therefore, by induction, Rn ; Rm = Rm+n holds for all n N

1. From (c), we have that if |S| = k, then Rk2 = Rk2+1

Assume that (a, b) Rk2 and (b, c) Rk2,

then (a, c) Rk2 ; Rk2

= R2k2 (from (d))

= Rk2 (from (a))

Therefore, Rk2 is transitive.

1. We need to prove the reflective, symmetric and transitive.

* Reflective: Since (*R* ∪ 𝑅←)0 = I, and from (b) we have that I (*R* ∪ 𝑅←)k2,

so for any a S, (a, a) I (*R* ∪ 𝑅←)k2.

Therefore, (*R* ∪ 𝑅←)k2 is transitive.

* Symmetric: Let p = (*R* ∪ 𝑅←)k2 be a binary relation.

Assume that (a, b) (*R* ∪ 𝑅←)k2 ∪ (*R* ∪ 𝑅← ; (*R* ∪ 𝑅←)k2)

From (c) we know that :

(a, c0) *R* ∪ 𝑅← and (c0, b) (*R* ∪ 𝑅←)k2-1

(c0, c1) *R* ∪ 𝑅←  and (c1, b) (*R* ∪ 𝑅←)k2-2

……

(ci, ci+1) *R* ∪ 𝑅←  and (ci+1, b) (*R* ∪ 𝑅←)k2

So for all i [0, k-1], (ci, ci+1) *R* ∪ 𝑅← (c0 = a, ck = b)

If (ci, ci+1) *R* ∪ 𝑅←, then (ci+1, ci) *R* ∪ 𝑅←. So (b, a) (*R* ∪ 𝑅←)k2

Therefore, (*R* ∪ 𝑅←)k2 is symmetric.

* Transitive: From (e) we have that, (*R* ∪ 𝑅←)k2 is transitive.

Therefore, if |S| = k, (*R* ∪ 𝑅←)k2  is an equivalence relation.



***Problem4***

If n = 1: f(1) O(1)

If n

Because ,

so f(n) f() + 3f() + n = 4f() + n = 4f() + O(1) O(n)

Therefore, f(n) O(n)

***Problem5***

1. count(T):

if T = τ:

return 0

else:

return count( Tleft ) + count( Tright ) + 1

1. leaves(T):

if T = τ:

return 0

else if T =((τ, τ)):

return 1

else:

return leaves( Tleft ) + leaves( Trignt )

1. half-leaves(T):

if T = τ:

return 0

else if T = (τ, Tright) or T = (Tletf, τ):

return 1

else:

return half-leaves( Tleft ) + half-leaves( Tright )

1. count(T) = 2\*leaves(T) + half-leaves(T) – 1

We prove P(T) by induction on T

Base case T = (τ, τ): count(T) = 1, leaves(T) = 1，half-leaves(T) = 0, so P((τ, τ)) holds.

Inductive case:T = (Tleft , Tright)

Assume P(Tleft) holds and P(Tright) holds, and Tleft , Tright are both not empty.

That is, count(Tleft) = 2 \* leaves(Tleft) + half-leaves(Tleft) – 1,

count(Tright) = 2 \* leaves(Tright) + half-leaves(Tright) – 1,

we will show that P((Tleft , Tright)) holds.

count(T) = count((Tleft , Tright ))

= count(Tleft ) + count(Tright) + 1 + half-leaves(Tright) +

half-leaves(Tleft) – 2

= 2\*leaves((Tleft , Tright)) + half-leaves((Tleft , Tright)) -1

= 2leaves(T) + half-leaves(T) – 1

So P(T) holds when Tleft and Tright are not empty.

Therefore, by the induction, P(T) holds for all binary trees T.

***Problem6***

1. O(n2)
2. O(n3)
3. recursive-product(A, B):

if n = 1:

AB = S\*W

if n = 2:

AB =

else:

recursive-product (SW)

recursive-product (TY)

recursive-product (SX)

recursive-product (TZ)

recursive-product (VW)

recursive-product (VY)

recursive-product (VX)

recursive-product (VZ)

We have:

T(1) = O(1)

T(n) = O(1) + T(logn) O(logn)

1. T(n) O(logn)