Homework 1

M1522.000900 Data Structure (2019 Fall)

2013-12815 Dongjoo Lee

**Question 1.**

(1) **a. Check reflective:**  1

∴ is reflective.

**b. Check symmetric:** if , then, .

1.

∴ is symmetric.

**c. Check transitive:** ifand, then,

[*case 1*] if is even, then is even and is even.

thus, is even --- (i)

[*case 2*] if is odd, then is odd and is odd.

thus, is even --- (ii)

From (i)&(ii),

∴ is transitive.

**∴ is an equivalence relation.**

(2) **a. Check reflective:**

∴ is not reflective.

**∴ is *not* an equivalence relation.**

(3) **a. Check reflective:** , thus,

∴ is reflective.

**b. Check symmetric:**

if , then,

, thus,

∴ is symmetric.

**c. Check transitive:** if and , then,

and

[*case 1*] if , then, and

thus, --- (i)

[*case 2*]if , then, and

thus, --- (ii)

From (i)&(ii),

∴ is transitive.

**∴ is an equivalence relation.**

(4) **a. Check reflective:**

∴ is reflective.

**b. Check symmetric:** if , then,

is not always integer, thus,

∴ is not symmetric

**∴ is *not* an equivalence relation.**

(5) **a. Check reflective:**

∴ is reflective.

**b. Check symmetric:** if , then, .

∴ is symmetric.

**c. Check transitive:** if and , then,

and

∴ is transitive.

**∴ is an equivalence relation.**

(6) **a. Check reflective:** thus

∴ is reflective.

**b. Check symmetric:** if , then,

thus

∴ is symmetric.

**c. Check transitive:** if and , then,

and

*[counter case]* when , then,

∴ is not transitive.

**∴ is *not* an equivalence relation.**

**Question 2.**

(1) **a. Check antisymmetric:** if , then, .

∴ is antisymmetric.

**b. Check transitive:**  if and , then,

is grandfather of , thus,

∴ is not transitive.

**∴ is *not* a partial ordering.**

(2) **a. Check antisymmetric:** if , then, .

∴ is antisymmetric.

**b. Check transitive:** if and , then,

is ancestor of , thus,

∴ is transitive.

**∴ is a partial ordering.**

(3) **a. Check antisymmetric:** if , then, .

∴ is antisymmetric.

**b. Check transitive:** if and , then,

is older than , thus,

∴ is transitive.

**∴ is a partial ordering.**

(4) **a. Check antisymmetric:** when is sister of , then, and , but .

∴ is not antisymmetric.

**∴ is *not* a partial ordering.**

(5) **a. Check antisymmetric:**  and , but, .

∴ is not antisymmetric.

**∴ is *not* a partial ordering.**

(6) **a. Check antisymmetric:**  there is no that and .

∴ is antisymmetric

**b. Check transitive:**  and and

∴ is transitive

**∴ is a partial ordering.**

**Question 3.**

(1) To move disks from L to R, there is process,

*a. move disks from L to M,*

*b. move disk from L to R,*

*c. move disks from M to R.*

And there’s no difference *from wherever to wherever* in terms of the number of movements.

So, we can define,

⋯ (i)

∴

(2) **Follows *(i)* above.**

(3) Expand the equation by using (i), with adding 1 to both sides,

Set , then, is geometric sequence.

by (ii), and equation holds when , therefore,

**∴**

**Question 4.**

(1)

From (i) to (iv), by multiply in both sides,

’

By summing up from (i)’ to (iv)’,

, and equation holds when , therefore,

(2) From given equation,

From closed-form solution, at

**∴ holds for**

(3) Assuming is true.

if , then,

**Question 5.**

**<Basis>**

∴  **holds for**

**<Inductive step>**

Assuming is true.

If , then,

∴  **also holds for**

**<Proof>**

By ,