RECURRENCE RELATIONS LP QUESTIONS	
LP5	Let F be the function such that F(n) is the sum of the first n positive integers. Give a recursive definition of F(n). **Step 1** 1 of 2 Given: $F(n)$ is the sum of the first n positive integers. $$F(n) = 1 + 2 + ... + (n-2) + (n-1) + n$$ $F(n-1)$ is then the sum of the first $n-1$ positive integers: $$F(n-1) = 1 + 2 + ... + (n-2) + (n-1)$$ We then note: $$F(n) = 1 + 2 + ... + (n-2) + (n-1) + n = F(n-1) + n$$ Next let us determine the first term. The first positive integer is 1 and thus the sum of the first 1 positive integers is then also 1: $$F(1) = 1$$ The recursive definition then becomes: $$F(1) = 1$$ $$F(n) = F(n-1) + n \text{ when } n \geq 2$$ **Result** 2 of 2 $$F(1) = 1$$ $$F(n) = F(n-1) + n \text{ when } n \geq 2$$
LP6	Give a recursive definition of the set of positive integers that are multiples of 5. **Solution** Verified **Step 1** 1 of 2 S is the set of positive integers that are multiples of 5. The first positive integer that is a multiple of 5 is 5: $$5 \in S$$ Every multiple of 5 is of the form $5k$ with k a positive integer, while $5(k-1)+5$, thus every multiple of 5 is the previous multiple of 5 increased by 5. $$s + 5 \in S \text{ whenever } s \in S$$ **Result** 2 of 2 $$5 \in S$$ $$s + 5 \in S \text{ whenever } s \in S$$

LP7

Give a recursive definition of the functions max and min so that max(a_1, a_2, . . . , a_n) and min(a_1, a_2, . . . , a_n) are the maximum and minimum of the n numbers a_1, a_2, . . . , a_n, respectively.

Solution Verified

Step 1

When $n = 1$, the maximum and minimum is the number itself:

$$\max(a_1) = a_1$$

$$\min(a_1) = a_1$$

When $n = 2$, then the maximum is defined as the larger value and the minimum is defined as the smaller value.

$$\max(a_1, a_2) = \begin{matrix} a_1 & \text{if } a_1 \geq a_2 \\ a_2 & \text{otherwise} \end{matrix}$$

$$\min(a_1, a_2) = \begin{matrix} a_2 & \text{if } a_1 \geq a_2 \\ a_1 & \text{otherwise} \end{matrix}$$

When $n > 2$, then the maximum of the n numbers is equal to the maximum of the maximum of the first $n - 1$ numbers and the nth number; while the minimum of the n numbers is equal to the minimum of the minimum of the first $n - 1$ numbers and the nth number

$$\max(a_1, a_2, ..., a_n, a_{n+1}) = \max(\max(a_1, a_2 ..., a_n), a_{n+1})$$

$$\min(a_1, a_2, ..., a_n, a_{n+1}) = \min(\min(a_1, a_2 ..., a_n), a_{n+1})$$

Result

$$\max(a_1, a_2, ..., a_n, a_{n+1}) = \max(\max(a_1, a_2 ..., a_n), a_{n+1})$$

$$\min(a_1, a_2, ..., a_n, a_{n+1}) = \min(\min(a_1, a_2 ..., a_n), a_{n+1})$$

LP8

Give a recursive definition of the function ones(s), which counts the number of ones in bit string s and analyse.

Step 1

1 of 3

(a) $ones(s)$ represents the number of ones in a bit string s and let S be the set of all bit strings.

Basis step We first define the number of ones of strings containing only 1 bit.

$$ones(0) = 0$$

$$ones(1) = 1$$

Recursive step Let $1s$ represent the string with the bit 1 added to the front of the string s, while $0s$ represents the string with the bit 0 added to the front of the string s.

$$ones(1s) = ones(s) + 1 \text{ whenever } s \in S$$

$$ones(0s) = ones(s) \text{ whenever } s \in S$$

Step 2

2

(b) To proof: $ones(st) = ones(s) + ones(t)$

PROOF BY STRUCTURAL INDUCTION

Basis step

Let s and t be 0 and/or 1

$$ones(00) = ones(0) + ones(0)$$

$$ones(10) = ones(1) + ones(0)$$

$$ones(01) = ones(0) + ones(1)$$

$$ones(11) = ones(1) + ones(1)$$

Thus the property is true for the basis step

Recursive step

$$ones(1s) = ones(s) + 1 = 1 + ones(s) = ones(1) + ones(s)$$

$$ones(0s) = ones(s) = 0 + ones(s) = ones(0) + ones(s)$$

Conclusion By the principle of structural induction, $ones(st) = ones(s) + ones(t)$

□

LP9

Give a recursive definition of the set of bit strings that are palindromes.

Step 1

Let S be the set of all bit strings that are palindrome.

Basis step The empty string is a palindrome and all strings containing only one bit is a palindrome.

$$\lambda \in S$$

$$0 \in S$$

$$1 \in S$$

Recursive step A string with more than 1 bit is a palindrome if the first and last letters are the same and the remaining string is also a palindrome.

$$0s0 \in S \text{ whenever } s \in S$$

$$1s1 \in S \text{ whenever } s \in S$$

Result

$$\lambda \in S$$

$$0 \in S$$

$$1 \in S$$

$$0s0 \in S \text{ whenever } s \in S$$

$$1s1 \in S \text{ whenever } s \in S$$

LP10.

Find these values of Ackermann's function. a) A(1, 0) b) A(0, 1) c) A(1, 1) d) A(2, 2)

Ackermann, is one of the simplest and earliest-discovered examples of a total computable function that is not primitive recursive. All primitive recursive functions are total and computable, but the Ackermann function illustrates that not all total computable functions are primitive recursive. Refer this for more.
It's a function with two arguments each of which can be assigned any non-negative integer.
Ackermann function is defined as:

$$A(m,n) = \begin{cases} n+1 & \text{if } m=0 \\ A(m-1,1) & \text{if } m>0 \text{ and } n=0 \\ A(m-1,A(m,n-1)) & \text{if } m>0 \text{ and } n>0 \end{cases}$$

where m and n are non-negative integers

Step 1

We use the definition of Ackermann's function. There are four cases in the definition depending on the values of (m, n).

$a)\ A(1, 0) = 0$

Explanation: In this case we have $m = 1$ and $n = 0$. So, the case $m \geq 1, n = 0$ applies.

$b)\ A(0, 1) = 2.$

Explanation: In this case we have $m = 0$, so $A(m, n) = 2n = 2(1) = 2.$

Step 2

$c)\ A(1, 1) = 2$

Explanation: In this case we have $m = 1$ and $n = 1$. So, the case $m \geq 1, n = 1$ applies for which the function takes value 2.

$d.\ A(2, 2) = 4.$

Explanation: We have $m = 2$ and $n = 2$. So, the case $m \geq 1, n \geq 2$ applies. We thus get

$$\begin{aligned} A(2,2) &= A(1, A(2,1)) \\ &= A(1,2) \\ &= A(0, A(1,1)) \\ &= 2A(1,1) \\ &= 2 \cdot 2 \\ &= 4 \end{aligned}$$

Result

(a)

$$ones(0) = 0$$

$$ones(1) = 1$$

$$ones(1s) = ones(s) + 1 \text{ whenever } s \in S$$

$$ones(0s) = ones(s) \text{ whenever } s \in S$$

(b) $ones(st) = ones(s) + ones(t)$

LP4

Given a positive integer n, list all the moves required in the Tower of Hanoi puzzle to move n disks from one peg to another according to the rules of the puzzle.

Tower of Hanoi using Recursion:
The idea is to use the helper node to reach the destination using recursion. Below is the pattern for this problem:

- Shift ‘N-1’ disks from ‘A’ to ‘B’, using C.
- Shift last disk from ‘A’ to ‘C’.
- Shift ‘N-1’ disks from ‘B’ to ‘C’, using A.

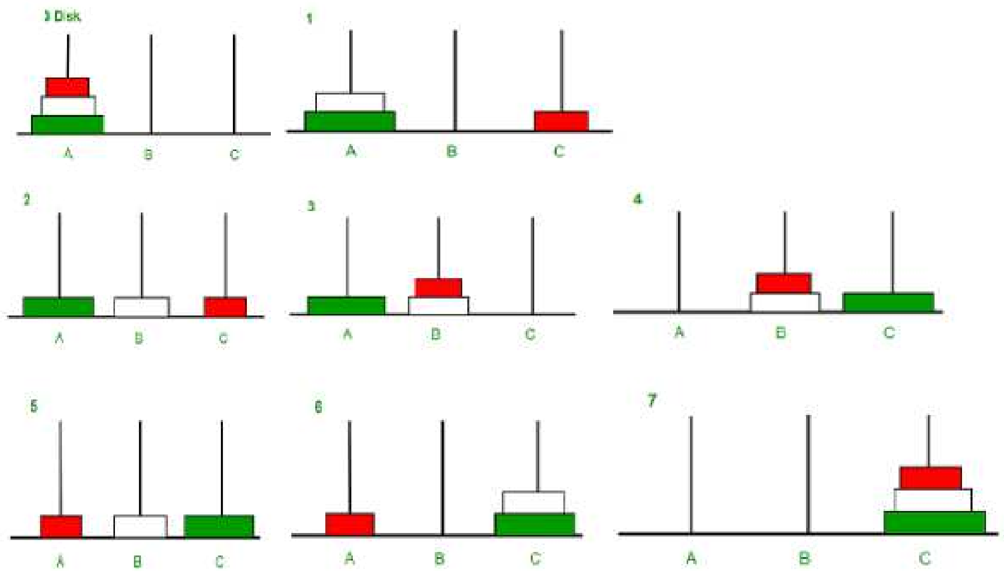

Image illustration for 3 disks

Move disk 1 from rod A to rod C

Move disk 2 from rod A to rod B

Move disk 1 from rod C to rod B

Move disk 3 from rod A to rod C

Move disk 1 from rod B to rod A

Move disk 2 from rod B to rod C

Move disk 1 from rod A to rod C