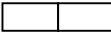


CHAPTER 8**ITERATION AND RECURSION**

- A loop invariant need not be true
 - at the start of the loop.
 - at the start of each iteration
 - at the end of each iteration
 - at the start of the algorithm**
- We wish to cover a chessboard with  dominoes, the number of black squares and the number of white squares covered by dominoes, respectively, placing a domino can be modeled by
 - $b := b + 2$
 - $w := w + 2$
 - $b, w := b+1, w+1$
 - $b := w$**
- If $m \times a + n \times b$ is an invariant for the assignment $a, b := a + 8, b + 7$, the values of m and n are
 - $m = 8, n = 7$
 - $m = 7, n = -8$**
 - $m = 7, n = 8$
 - $m = 8, n = -7$
- Which of the following is not an invariant of the assignment?
 $m, n := m+2, n+3$
 - $m \bmod 2$
 - $n \bmod 3$
 - $3 \times m - 2 \times n$
 - $2 \times m - 3 \times n$**
- If Fibonacci number is defined recursively as

$$F(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F(n-1) + F(n-2) & \text{otherwise} \end{cases}$$
 to evaluate $F(4)$, how many times $F()$ is applied?
 - 3**
 - 4
 - 8
 - 9
- Using this recursive definition

$$a^n = \begin{cases} 1 & \text{if } n = 0 \\ a \times a^{n-1} & \text{otherwise} \end{cases}$$
 how many multiplications are needed to calculate a^{10} ?
 - 11
 - 10
 - 9**
 - 8

Part II**1. What is an invariant?**

- An expression involving variables, which remains unchanged by an assignment to one of these variables, is called an invariant of the assignment.
- An expression of the variables has the same value before and after an assignment, it is an invariant of the assignment.

2. Define a loop invariant.

- An invariant the loop body is known as a loop invariant.
- When the loop ends, the loop invariant has the same value.

3. Does testing the loop condition affect the loop invariant? Why?

Yes, it affects. When a loop ends, the loop invariant is true. In addition, the termination condition is also true.

4. What is the relationship between loop invariant, loop condition and the input output recursively?

LOOP INVARIANT	LOOP CONDITION
An invariant for the loop body is known as a loop invariant.	An loop condition that produce the result based particular condition.
A loop invariant is a condition that is necessarily true immediately before and immediately after each iteration of a loop.	A loop invariant is some condition that holds for every iteration of the loop.

5. What is recursive problem solving?

Each solver should test the size of the input. If the size is small enough, the solver should output the solution to the problem directly. If the size is not small enough, the solver should reduce the size of the input and call a sub-solver to solve the problem with the reduced input.

6. Define factorial of a natural number recursively.

“The **factorial** of a **number** is the product of all the integers from 1 to that **number**.”

For example, the **factorial** of 4 (denoted as 4!) is $1*2*3*4=24$.

Factorial (4)

i = 1, f = 1;

= f = 1 x 1 = 1

= f = 1 x 2 = 2

= f = 2 x 3 = 6

= f = 6 x 4 = 24

7. What is Iteration?

In iteration, the loop body is repeatedly executed as long as the loop condition is true. Each time the loop body is executed, the variables are updated. However, there is also a property of the variables which remains

8. Define :Recursion

Recursion is another algorithm design technique, closely related to iteration, but more powerful. Using recursion, we solve a problem with a given input, by solving the same problem with a part of the input, and constructing a solution to the original problem from the solution to the partial input.

9. Write the recursive process with solvers for calculating power(2, 5)

power (2,5)

= 2 × power (2,4)

= 2 × 2 × power(2,3)

= 2 × 2 × 2 × power(2, 2)

= 2 × 2 × 2 × 2 × power (2,1)

= 2 × 2 × 2 × 2 × 2 × power (2,0)

= 2 × 2 × 2 × 2 × 2 × 1

= 2 × 2 × 2 × 2 × 2

= 2 × 2 × 2 × 4

= 2 × 2 × 8

= 2 × 16

= 32

Part III

1. There are 7 tumblers on a table, all standing upside down. You are allowed to turn any 2 tumblers simultaneously in one move. Is it possible to reach a situation when all the tumblers are right side up? (Hint: The parity of the number of upside down tumblers is invariant.)

Solution:

This Tumbler problem Result (Output) is needs to be turn all tumblers UP.

[u – no of tumblers]

1. Two tumblers can be both upside up. After turning u – increments by 2.
2. Two tumblers are both upside down. After turning u – decrements by 2.
3. One is upside down and another is proper.

U – is not changed.

So, after every step.

U – is either incremented by 2 or decremented 2 or kept the same.

We can ignore the condition of u – not being changed.

Now, $u := u + 2$ (or) $u := u - 2$

The invariant in this is that parity of u – is not changing. i.e; if u – is even at the beginning, its not changed at all and similarly if u – is odd.

That invariant is the initial state that needs to be defined. The final requirement is that u-needs to become zero. This is possible only when the parity of u-is zero, u is even.

The final solution is if the number of tumblers that are upside down is even it is possible to get to a state by repeating the process for all the tumblers that are upside up.

2. A knockout tournament is a series of games. Two players compete in each game; the loser is knocked out (i.e. does not play any more), the winner carries on. The winner of the tournament is the player that is left after all other players have been knocked out.

Suppose there are 1234 players in a tournament. How many games are played before the tournament winner is decided?

Solution:

Knockout tournament

1 2 3 4 initial number of players

each round a player is laid off thus

n. number of remaining players.

r - total number of rounds

k - number of rounds held assuming n is even

$n, r := n - k, r + k$

$n + r$ is invariant

winners is decided when $n = 1$

initially $n + r = 1234$, therefore $r = 1233$! (makes sense = 1233 rounds will eliminate 1233 players).

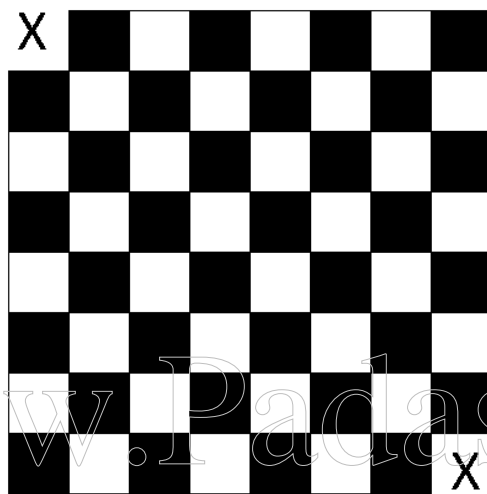
3. King Vikramaditya has two magic swords. With one, he can cut off 19 heads of a dragon, but after that the dragon grows 13 heads. With the other sword, he can cut off 7 heads, but 22 new heads grow. If all heads are cut off, the dragon dies. If the dragon has originally 1000 heads, can it ever die? (Hint: The number of heads mod 3 is invariant.)

Solution:

- If it is 1000 heads:
- You can now cut off 981 heads (multiple of 3) then cut the last 19 and the dragon will die.
- The 13 heads won't come back after the dragon is killed. Unless it's a magic dragon.

Part IV

1. Assume an 8×8 chessboard with the usual coloring. "Recoloring" operation changes the color of all squares of a row or a column. You can recolor repeatedly. The goal is to attain just one black square. Show that you cannot achieve the goal. (Hint: If a row or column has b black squares, it changes by $(|8 - b) - b|$).

Solution:**2. Power can also be defined recursively as**

$$a^n = \begin{cases} 1 & \text{if } n = 0 \\ a \times a^{n-1} & \text{if } n \text{ is odd} \\ a^{n/2} \times a^{n/2} & \text{if } n \text{ is even} \end{cases}$$

Construct a recursive algorithm using this definition. How many multiplications are needed to calculate a^{10} ?

Solution:

Power (a, n)

Input n - is an integer, $n \geq 0$

-- Output : a^n

if $n = 0$ – base case

1

else

if $(n \% 2 != 0)$ – recursion step in case of odd

$a \times \text{power}(a, n - 1)$

else

$a \times \text{power}(a, n/2)$ --- recursion step in case of even.

3. A single-square-covered board is a board of $2n \times 2n$ squares in which one square is covered with a single square tile. Show that it is possible to cover the this board with triominoes without overlap.

Solution:

$$2^{2n} - 1$$

To identify the 2-by-2 square containing the missing square; and we place our first L-shaped piece to finish covering that 2-by-2 square:

```

O O O O O O O O
O O O O O O O O
O O O O O O a a
O O O O O O x a
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O

```

www.Padasalai.Net

M.VIJAYA KUMAR, MCA.,M.Phil.,B.Ed.,PGDCA.,
PGT-COMPUTER TEACHER,

V.SUJITHA, MCA.,M.Phil.,
PGT-COMPUTER TEACHER,

SASTRA MATRICULATION HIGHER SECONDARY SCHOOL,
KILPENNATHUR, TIRUVANNAMALAI 604601

Mobile No : 9655826843

Email : vijay28soft@gmail.com

*****ALL THE BEST*****

