Assignment 3

Name: Bharat Upadhyay
2017641
PCS 302(Data Structures with C)
Section : AI & DS

Q1 – What is the difference between recursion and iteration?

Ans:

Recursion:

When a function calls itself within its own code, it is termed as Recursion. If the recursive function does not meet to a termination condition, it leads to an infinite recursion. There is a chance of system crash in infinite recursion. It is comparatively slower than iteration. It utilizes the stack for calling.

Example:

To find the factorial of a number using Recursion :

```
#include <stdio.h>
int fact(int n)
{
   if (n == 0)
      return 1;
   else
      return n * fact(n-1);
}

int main()
{
   int n,f;
   printf("Enter any number: ");
   scanf("%d", &n);
   f = fact(n);
   printf("factorial = %d",f);
}
```

Iteration:

In iteration, there is a repeated execution of the set of instructions. In Iteration, loops are used to execute the set of instructions repetitively until the condition is false. It is comparatively faster than recursion. It does not utilize stack.

Example:

```
#include <stdio.h>
int fact(int num)
{
   int res = 1, i;

    for (i = 2; i <= num; i++)
       res *= i;
    return res;
}

int main()
{
    int num, f;
    printf("Enter any number: ");
    scanf("%d", &num);
    f = fact(num);
    printf("factorial = %d", f);
}
```

Q2 – What is tail recursion?

Ans:

The tail recursion is basically using the recursive function as the last statement of the function. So when nothing is left to do after coming back from the recursive call, that is called tail recursion.

Example:

```
// An example of tail recursive function
void print(int n)
{
   if (n < 0)
     return;
   printf("%d ",n);

   // The last executed statement is recursive call
   print(n-1);
}
```

Q3 – Define recursion tree. What are the uses of recursion tree? What can you conclude from this?

Ans:

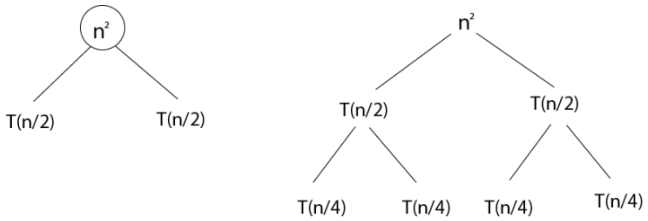
Definition :

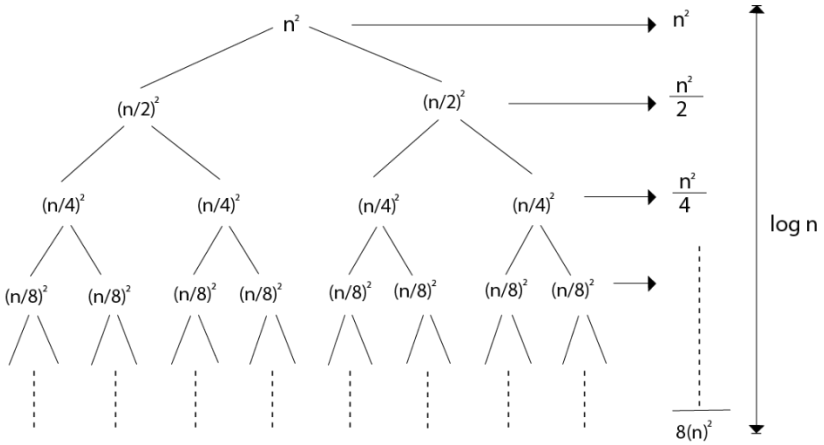
It is a method to analyse the complexity of an algorithm by diagramming the recursive function calls. Recursion Tree Method is a pictorial representation of an iteration method which is in the form of a tree where at each level nodes are expanded. We sum the costs within each of the levels of the tree to obtain a set of pre-level costs and then sum all pre-level costs to determine the total cost of all levels of the recursion. A Recursion Tree is best used to generate a good guess, which can be verified by the Substitution Method.

Steps:

- Draw a recursion tree based on the given recurrence relation.
- **Determine**: Cost of each level. Total number of levels in the recursion tree, Number of nodes in the last level. Cost of the last level.
- Add cost of all the levels of the recursion tree and simplify the expression so obtained in terms of asymptotic notation.

Example: Consider T (n) = $2T\left(\frac{n}{2}\right) + n^2$





$$T(n) = n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \ldots\ldots \log n \text{ times.}$$

$$\leq n^2 \sum_{i=0}^{\infty} \left(\frac{1}{2^i}\right)$$

$$\leq n^2 \left(\frac{1}{1-\frac{1}{2}}\right) \leq 2n^2$$

$$\mathbf{T(n) = \theta n^2}$$