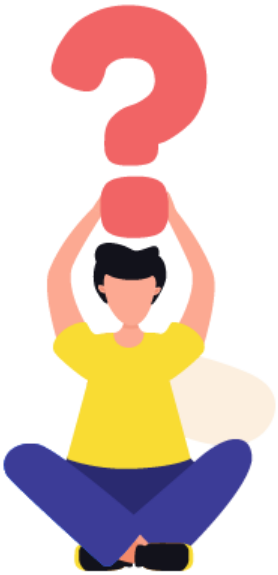
The Different Types of recursion

Many types of recursion can be presented in algorithms, it depends on how, where and how many call we have in our solution. We can cite the following type :

1. Terminal recursion: the function ends with the single recursive call.
2. Multiple recursion: if one of the treated cases is resolved with several recursive calls.
3. Cross or mutual recursion: two algorithms are mutually recursive if one uses the other and vice versa.
4. Nested recursion: if the function contains a call to itself as a parameter.

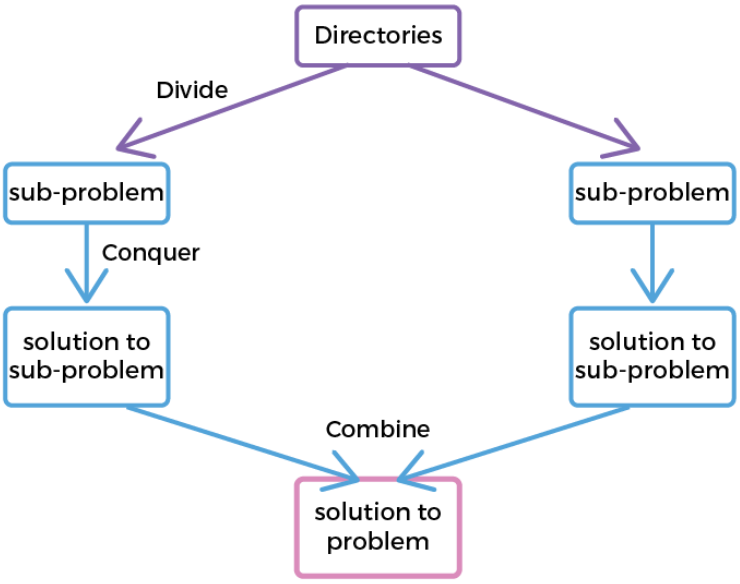
Do not worry we will see some example of each type in next.



The Different types of recursion

In addition, some algorithmic paradigm (techniques of solving problems) use recursion, the most popular is Divide and Conquer which solves a problem using following three steps:

- Divide: Break the given problem into subproblems of same type and same size.
- Conquer: Recursively solve these subproblems.
- Combine: Appropriately combine the answers.



We see some examples of this type of paradigm in the previous SuperSkill, they are Merge Sort, Quick Sort, Binary Sort.

Recursive Algorithm

The first example we are going to see is the sum of elements of an array.

We have iteratively seen this algorithm. Now, we are going to see the recursive version.

```
FUNCTION linearSum(arr : ARRAY_OF INTEGER, n :INTEGER) : INTEGER
BEGIN
   IF (n = 1) THEN // when the n is the length of the arr
      RETURN arr[0];
   ELSE
       // here we recall the function with the length - 1
       RETURN arr[n-1] + linearSum(arr,n-1);
   END_IF
END
```

The second example is going to be the famous fibonacci sequence in a recursive way.

```
FUNCTION fibbonacci(n:INTEGER) : INTEGER
IF (n=0) THEN
   RETURN 0;
ELSE_IF (n=1) THEN
   RETURN 1
ELSE
```

```
    RETURN fibbonacci(n-1) + fibbonacci(n-2)
END_IF
END
```

Our third example is going to be the Hofstadter sequence.

In this example, we are going to be interested on Hofstadter Female and Male sequences.

```
/*
In mathematics, a Hofstadter sequence is a member of
a family related integer sequences defined by non-linear
recurrence relations.
*/

/*
F (0) = 1
M (0) = 0
F (n) = n-M(F(n - 1)), n > 0
M (n) = n-F(M(n-1)), n>0
*/
FUNCTION hofstader_female(n : INTEGER) : INTEGER
BEGIN
   IF (n = 0) THEN
       RETURN 1
   ELSE
   RETURN n - hofstader_male(hofstader_female(n-1)) ;
   END_IF
END
FUNCTION hofstader_male(n : INTEGER) : INTEGER
BEGIN
   IF (n = 0) THEN
       RETURN 0;
   ELSE
   RETURN n - hofstader_female(hofstader_male(n-1)) ;
```

```
    END_IF
END
```

Our final example will the binary sum which is the divide and conquer version of our first example.

```
FUNCTION binarySum(arr : ARRAY_OF INTEGER,i,n : INTEGER) : INTEGER
BEGIN
    IF (n = 1) THEN
        RETURN arr[i]
    ELSE
        RETURN binarySum(arr,i,n/2)+binarySum(arr,i+n/2,n/2) ;
    END_IF
END
```

Previous next