



A Recursion Function

Report Issue (<https://github.com/LeetCode-Feedback/LeetCode-Feedback/issues>)

For a problem, if there exists a recursive solution, we can follow the guidelines below to implement it.

For instance, we define the problem as the function $F(X)$ to implement, where X is the input of the function which also defines the scope of the problem.

Then, in the function $F(X)$, we will:

1. Break the problem down into smaller scopes, such as $x_0 \in X, x_1 \in X, \dots, x_n \in X$;
2. Call function $F(x_0), F(x_1), \dots, F(x_n)$ **recursively** to solve the subproblems of X ;
3. Finally, process the results from the recursive function calls to solve the problem corresponding to X .

Example

To showcase the above guidelines, we give another example on how to solve a problem recursively.

Given a linked list, swap every two adjacent nodes and return its head.

e.g. for a list 1-> 2 -> 3 -> 4, one should return the head of list as 2 -> 1 -> 4 -> 3.

We define the function to implement as `swap(head)`, where the input parameter `head` refers to the head of a linked list. The function should return the `head` of the new linked list that has any adjacent nodes swapped.

Following the guidelines we lay out above, we can implement the function as follows:

1. First, we swap the first two nodes in the list, i.e. `head` and `head.next`;
2. Then, we call the function self as `swap(head.next.next)` to swap the rest of the list following the first two nodes.
3. Finally, we attach the returned head of the sub-list in step (2) with the two nodes swapped in step (1) to form a new linked list.

As an exercise, you can try to implement the solution using the steps we provided above. Click on "Swap Nodes in Pairs" to try to implement the solution yourself.