二分查找代码模板

```
# Python
left, right = 0, len(array) - 1
while left <= right:
      mid = (left + right) / 2
      if array[mid] == target:
            # find the target!!
            break or return result
      elif array[mid] < target:
            left = mid + 1
      else:
            right = mid - 1
```

DFS代码模板

递归写法

```
visited = set()

def dfs(node, visited):
    if node in visited: # terminator
        # already visited
        return

    visited.add(node)

    # process current node here.
    ...
    for next_node in node.children():
        if next_node not in visited:
            dfs(next_node, visited)
```

非递归写法

```
def DFS(self, tree):

    if tree.root is None:
        return []

    visited, stack = [], [tree.root]

    while stack:
        node = stack.pop()
        visited.add(node)

        process (node)
        nodes = generate_related_nodes(node)
        stack.push(nodes)

    # other processing work
    ...
```

BFS代码模板

```
# Python
def BFS(graph, start, end):
    visited = set()
    queue = []
    queue.append([start])

    while queue:
        node = queue.pop()
        visited.add(node)

        process(node)
        nodes = generate_related_nodes(node)
        queue.push(nodes)

    # other processing work
    ...
```