

A BFS - Template

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

Previously, we have already introduced two main scenarios of using BFS: do traversal or find the shortest path. Typically, it happens in a tree or a graph. As we mentioned in the chapter description, BFS can also be used in more abstract scenarios.

In this article, we will provide you with a template. Then, we provide some exercise after this article for practice.

It will be important to determine the nodes and the edges before doing BFS in a specific question. Typically, the node will be an actual node or a status while the edge will be an actual edge or a possible transition.

Template I

Here we provide a pseudocode for you as a template:

Java

📄 Copy

```

2  * Return the length of the shortest path between root and target node.
3  */
4  int BFS(Node root, Node target) {
5      Queue<Node> queue; // store all nodes which are waiting to be processed
6      int step = 0;      // number of steps needed from root to current node
7      // initialize
8      add root to queue;
9      // BFS
10     while (queue is not empty) {
11         // iterate the nodes which are already in the queue
12         int size = queue.size();
13         for (int i = 0; i < size; ++i) {
14             Node cur = the first node in queue;
15             return step if cur is target;
16             for (Node next : the neighbors of cur) {
17                 add next to queue;
18             }
19             remove the first node from queue;
20         }
21         step = step + 1;
22     }
23     return -1; // there is no path from root to target
24 }
```

1. As shown in the code, in each round, the nodes in the queue are the nodes which are waiting to be processed.
2. After each outer while loop, we are one step farther from the root node. The variable step indicates the distance from the root node and the current node we are visiting.

Template II

Sometimes, it is important to make sure that we never visit a node twice. Otherwise, we might get stuck in an infinite loop, e.g. in graph with cycle. If so, we can add a hash set to the code above to solve this problem. Here is the pseudocode after modification: