**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 exercises 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:

A screenshot of a computer code

AI-generated content may be incorrect.

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:

A screenshot of a computer code

AI-generated content may be incorrect.

There are some cases where one does not need keep the visited hash set:

1. You are absolutely sure there is no cycle, for example, in tree traversal;
2. You do want to add the node to the queue multiple times.