

什么是队列（Queue）？

队列（queue）是一种采用先进先出（FIFO，first in first out）策略的抽象数据结构，最常用的就是在**宽度优先搜索(BFS)**中，记录待扩展的节点。

队列内部存储元素的方式，一般有两种，数组（array）和链表（linked list）。两者的最主要区别是：

- 数组对随机访问有较好性能。
- 链表对插入和删除元素有较好性能。

Java常用的队列包括如下几种：

ArrayDeque：数组存储。实现Deque接口，而Deque是Queue接口的子接口，代表**双端队列**（double-ended queue）。

LinkedList：链表存储。实现List接口和Duque接口，不仅可做队列，还可以作为双端队列，或栈（stack）来使用。

面试中常用的interface种类：

Set

- **HashSet**
 - 无重复数据
 - 可以有空数据
 - 数据无序

```
Set<String> set = new HashSet<>();
for (int i = 1; i < 6; i++) {
    set.add(i + "");
}
set.add("1"); //不会重复写入数据
set.add(null);//可以写入空数据
Iterator<String> iter = set.iterator();
while (iter.hasNext()) {
    system.out.print(iter.next() + " "); //数据无序
} // 输出(无序)为 3 4 1 5 null 2
```

- **TreeSet**
 - 无重复数据
 - 不能有空数据
 - 数据有序

```
Set<String> set = new TreeSet<>();
for (int i = 1; i < 6; i++) {
    set.add(i + "");
}
set.add("1"); //不会重复写入数据
//set.add(null);//不可以写入空数据
Iterator<String> iter = set.iterator();
while (iter.hasNext()) {
    system.out.print(iter.next() + " "); //数据有序
} // 输出(有序)为 1 2 3 4 5
```

Map

- **HashMap**
 - key 无重复，value 允许重复
 - 允许 key 和 value 为空
 - 数据无序
- **TreeMap**
 - key 无重复，value 允许重复
 - 不允许有null
 - 有序(存入元素的时候对元素进行自动排序，迭代输出的时候就按排序顺序输出)

List

一个 List 是一个元素有序的、可以重复(这一点与Set和Map不同)、可以为 null 的集合，List的实现类在面试中常用是：**LinkedList** 和 **ArrayList**

- **LinkedList**
 - 基于链表实现
- **ArrayList**
 - 基于动态数组实现
- **LinkedList** 与 **ArrayList** 对比：
 - 对于随机访问get和set，ArrayList绝对优于LinkedList，因为LinkedList要移动指针
 - 对于新增和删除操作add和remove，LinedList比较占优势，因为ArrayList要移动数据

什么时候使用BFS？

细一点的划分的话，这一类的问题还可以分为：

- 层级遍历 Level Order Traversal
- 由点及面 Connected Component
- 拓扑排序 Topological Sorting

典型问题：

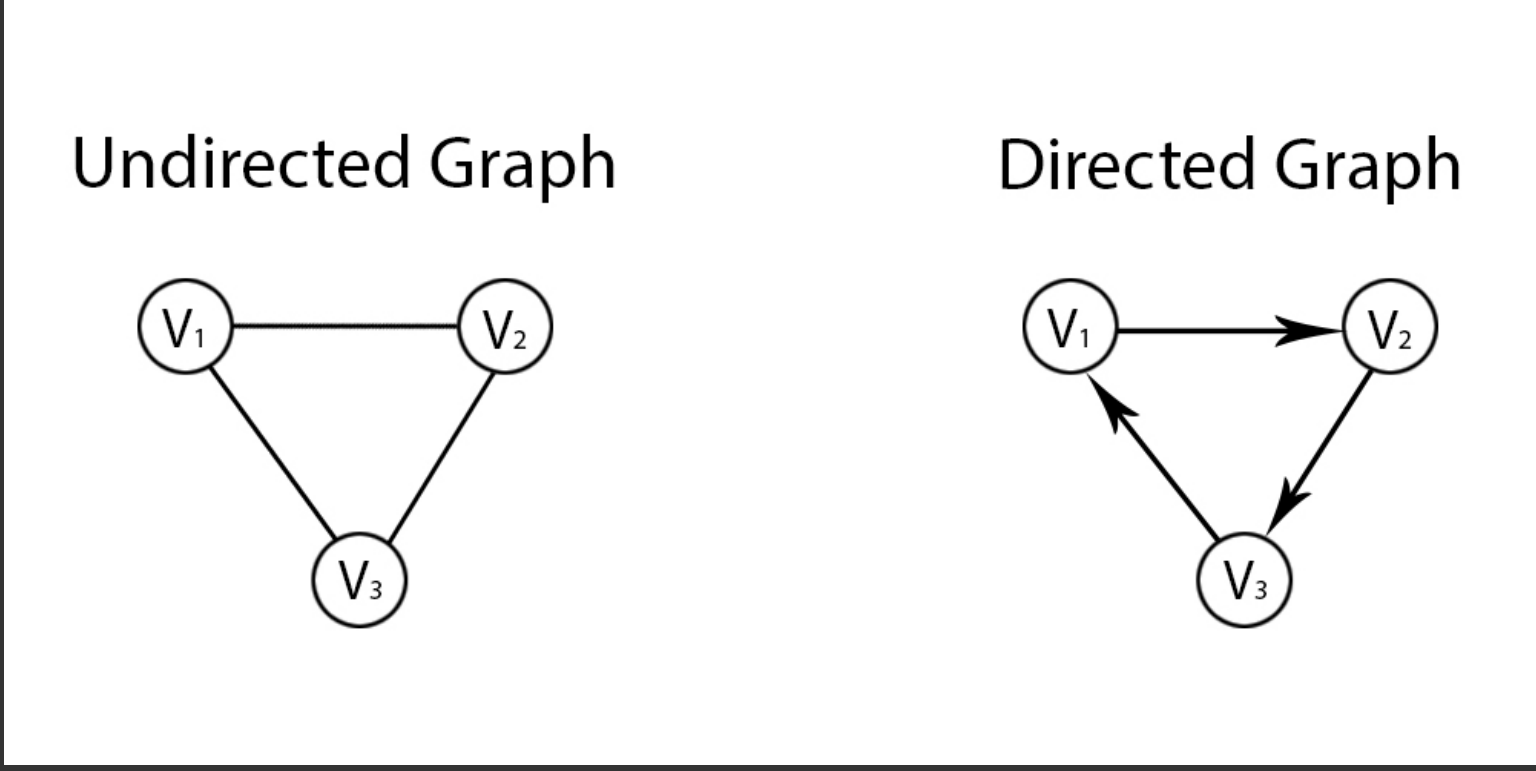
最短路径 Shortest Path in Simple Graph

什么是图（Graph）

图在离线数据中的表示方法为 <E，V>，E表示 Edge，V 表示 Vertex。也就是说，图是顶点（Vertex）和边（Edge）的集合。

图分为：

- 有向图（Directed Graph）
- 无向图（Undirected Graph）



(树也是特殊的图)

二叉树的BFS和图的BFS

二叉树中进行 BFS 和图中进行 BFS 最大的区别就是二叉树中无需使用 HashSet（C++: unordered_map, Python: dict）来存储访问过的节点（丢进过 queue 里的节点）
因为二叉树这种数据结构，上下层关系分明，没有环（circle），所以不可能出现一个节点的儿子是自己的情况。
但是在图中，一个节点的邻居的邻居可能就是自己了。

图的数据结构定义

有很多方法可以存储一个图，最常用的莫过于：

1. 邻接矩阵
2. 邻接表

而邻接矩阵因为耗费空间过大，我们通常在工程中都是使用邻接表作为图的存储结构。