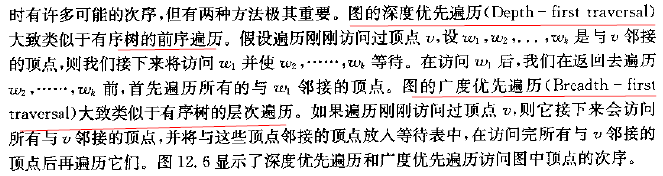
## 书写此文档的目的：

深度优先和过度优先对于学习计算机的人来说是熟悉的，无论是在人工智能的搜索算法中，还是在算法设计回溯算法的设计中，深度和广度是它们的基础，因此系统的分析图按照深度原则或者广度原则如何进行操作和程序实现这些操作的具体细节对于我来说是至关重要的。

## 图的深度优先搜索



上述文档详细论述了图遍历的次序。知道了如何进行操作，接下来的任务就是写出程序让计算机按照我们的原则来进行操作。现在我需要的

## 算法设计

注意图深度优先算法一般写出两层的形式，第二层是一个递归函数，第一层则负责启动递归，递归一旦启动，就像多米诺骨牌一般一直运行下去直到终点

下面程序是核心算法

1. template <int graph\_size, class Weight>
2. void Digraph<graph\_size, Weight>::traverse(Vertex v, bool visited[], void(\*visite)(Vertex &))const
3. {
4. visite[v] = true;
5. (\*visite)(v); //和树的前序遍历类似，一旦遇到结点，就首先访问该结点
6. for (Vertex w = 0; w < count; w++){
7. if ((adjacency[v][w] != false )&& (!visite[w]))
8. traverse(w, visisted, visit);
9. }
10. }