

# VORTEZ WOHL

---

**作者：**吴子豪

**邮箱：**[vortezwohl@proton.me](mailto:vortezwohl@proton.me)

**语言兼容：**C++98 及其后续标准

**命名空间：**namespace wohl;

**已包含：**"stdio.h", "string.h", "iostream"

---

## 类:

**Class Graph.**

//基于邻接链表，用以描述有向图的对象

**Class GStack.**

//基于整型数，用于实现栈的对象

**Class GQueue.**

//基于整形数，用于实现队列的对象

---

## 在类型 Graph 中:

一共有 14 个成员方法

**Graph\* setv(int idx);**

//添加一个新顶点

**Graph\* setv(int\* idxs);**

//根据形参处提供的整形数数组，添加一组顶点，并按照大小顺序排列

//数据最后一个元素必须是 END 或者 MAXINT，以提示该函数读取终点

**Graph\* sete(int idx1,int idx2,int weit);**

//创建一条从点 idx1 指向点 idx2 的边

**Graph\* rmv(int idx);**

//删去一个既有顶点

**Graph\*\* edgeof(int idx);**

//返回一个存储了该点的边的主存地址序列和边的权重序列的数组

//最好和 print\_adjv(int idx,Graph\* vert)一起使用，以输出边的相关数据

**Graph\* locate(int idx);**

//返回当前节点在节点链表中的主存地址

**Graph\* locate(int idx1,int idx2);**

//返回当前节点在点 idx 的边链表中的主存地址

**void init\_traversal\_status();**

//初始化各节点被遍历状态，如果不初始化，那么遍历算法将无运行结果

**friend void traverse\_around(Graph& graph,Graph\* target);**

//向命令行终端打印该点和该点的所有邻接点的序号

**friend void greedy\_traversal(Graph& graph,Graph\* target);**

//用类似贪心的方式遍历该有向图，并且把结果输出给命令行终端

**friend void DFS(Graph& graph,Graph\* target);**

//深度优先搜索，并把结果输出给命令行终端

**void print\_Graphes();**

//显示当前所有节点的序号

**void print\_adjv(int idx,Graph\* vert)**

//打印点 idx 的一条邻接边，最好和 edgeof(int idx)一起使用

//因为 edgeof(int idx)将直接返回一个 Graph 类的数组，可以在主程序中用循环遍历数组的方式访问邻接边的信息

---

## 在类型 GStack 中:

一共有两个成员方法

//进栈，出栈操作

**int g\_push(int idx);**

**int g\_pop();**

---

## 在类型 GQueue 中:

一共有两个成员方法

//入队，出队操作

**int g\_enqueue(int idx);**

**int g\_dequeue();**

---

