

邻接表存储和遍历的标准模板

Ahub_Ha 于 2020-09-04 12:10:11 发布

邻接表声明

```
1 int h[N], e[M], w[M], ne[M], idx; //邻接表 N节点个数, M边个数
2
3 其中
4 h[a] 指向a节点起点的邻接表列表的最后一个元素。
5 e[idx] 为当前idx编号的边指向的终点节点
6 w[idx] 为当前idx编号的边权重
7 ne 存储邻接表链表, 当前值对应邻接表中下一个的地址, 类似于值是指针。
8
```

邻接表构建

```
1 void add(int a, int b, int c) { //构建邻接表
2     e[idx] = b;
3     w[idx] = c;
4     ne[idx] = h[a];
5     h[a] = idx++; //h[a] 一直指向a邻接表头插法起点, 其实是最后一个, 指针保留的方式也是向前
6 }
7 1. idx一直向前, 如果a是第一次出现, 则h[a]的值对应ne中位置即是起点。
8 2. 插入的方式是类似头插法, 每次邻接表中的新元素出现, 则插入邻接链表的第一个。也可以这样理解, 是每次插到最后, 让h[a]指向最后一个元素, 遍历的时候倒序
9 3. 如果指向下一个为空时, 指针值为-1.
```

邻接表遍历

```
1 for(int i = h[vel]; i != -1; i = ne[i]) {
2     //TODO
3 }
4
5 i = ne[i] 模拟链表指针的next操作
6 h[vel] 指向vel链表的最后一个, 遍历是从后往前的
```

文章知识点与官方知识档案匹配，可进一步学习相关知识

算法技能树 leetcode-链表 82-删除排序链表中的重复元素 II 13051 人正在系统学习中