

第7章 结构体和文件

第7.7课 节点的定义

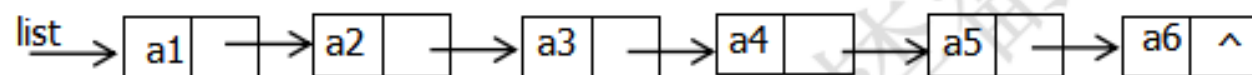
《信息学奥赛一本通·编程启蒙 C++版》

一、链表

用一组地址任意的存储单元(可以连续,也可不连续)依次存储线性表中的各个元素。链表可以用指针来实现,也可以用数组来实现。

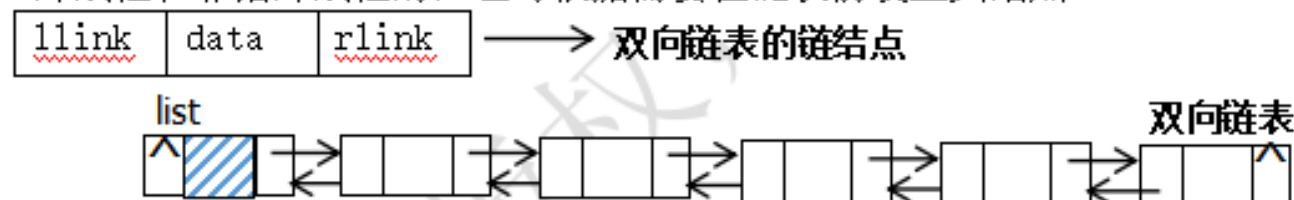
1. 单向链表/线性链表

此链表中每个节点链结点由两部分构成:元素自身信息即数据域(用 data 表示)、指示其直接后继元素位置的信息即指针域(用 link 表示)。整个链表由一个称为外指针/头结点指针 list 指出,以表明链表的首地址,当链表为空时, list 为 null。用线性链表存储线性表时,数据元素间的逻辑关系通过指针反映出来。



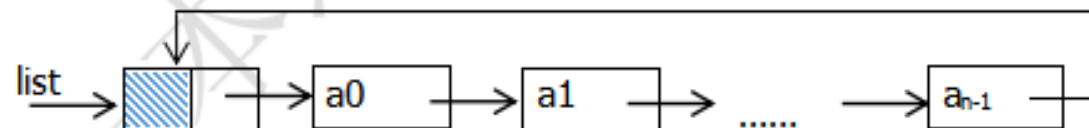
2. 双向链表

双向链表的每个链结点除了数据域 data 外设置两个指针域,一个 llink 指向直接前驱结点,一个 rlink 指向直接后继结点。双向链表有循环线性和非循环线性的,也可根据需要在链表前设置头结点 list。



3. 循环链表

链表的最后一个链结点的指针指向链表的第 1 个链结点,整个链表形成一个环,从表中任意节点出发均可找到表中其他节点。



三、用 struct 和指针定义节点

每个节点一般包括两个部分，一个是数据域，另一个是指针域。指针域用来指向和它本身一样的另一个结构体，也就是另一个节点。下面是数据域是整型的单向链表的节点的定义方式。

```
1. // 节点的定义
2. struct node{
3.     int data; //数据域
4.     struct node*next; //指针域
5. }NODE,*PNODE;
```

如果要定义二叉树的节点，第四行代码的指针域可以改成：

```
struct node*leftChild;
struct node*rightChild;
```

【例 77.1】模拟链表

【题目描述】

在图论题编程中，通常要运用邻接链表数据结构。由于动态指针比静态的数组的存取慢，很多 OI 选手就用数组模拟指针。现在就来学习一下这种方法的编程。

有 N 个点，编号从 1 到 N 。有 M 条边，每条边用连接的 2 个顶点表示，如：(3, 8)，表示顶点 3 和 8 之间的边（无向边）。请输出每个顶点通过边相邻的顶点。

【输入格式】

第 1 行， N 和 M 两个整数， N 范围在 $[1 \cdots 5000]$ ， M 范围在 $[1 \cdots 100000]$ ；下面有 M 行，每行两个整数，表示一条边。

【输出格式】

N 行，第 i 行的第 1 个数 k 表示有多少边和 i 号顶点相连，后面有 k 个数，表示哪 k 个顶点和 i 连接为一条边

【样例输入】

```
5 6
1 3
2 4
1 4
2 3
3 5
2 5
```

【样例输出】

```
2 4 3
3 5 3 4
3 5 2 1
2 1 2
2 2 3
```

【代码实现】

```
1. #include<iostream>
2. using namespace std;
3. struct node{
4.     int v;
5.     int next;
6. };                                //定义 struct 类型
7. node a[200001];                  //无向图，空间是边的两倍
8. int n,m,p;                       //p 为a 数组的空余空间下标
9. int k[5001],c[5001];             //邻接链表的表头，k 数组记长度
10. void insert(int u,int v) //把v 点插入到u 点的邻接链表前
11. {
12.     a[++p].v=v;                  //申请一个新节点
13.     a[p].next=c[u];
14.     c[u]=p;                     //插入到u 链表头
15.     k[u]++;                     //链表长度增加
16. }
17. int main(){
18.     cin>>n>>m;
19.     for (int i=1; i<=m; i++){
20.         int u,v;
```

```
21.         cin>>u>>v;
22.         insert(u,v);           //插入 v 点到链表 u
23.         insert(v,u);           //插入 u 点到链表 v
24.     }
25.     for (int i=1; i<=n; i++){
26.         cout<<k[i]<<' ';        //表的长度
27.         for (int j=c[i];j>0;j=a[j].next) //遍历链表
28.             cout<<a[j].v<<' ';
29.         cout<<endl;
30.     }
31.     return 0;
32. }
```

【例 77.2】经验值

【题目描述】

小明在玩一个游戏，游戏是这样的，有一个 1 到 n 的排列，我们按照某种给定顺序依次删除每一个数，每删除一个数，小明将会获得这个数左边的数加上这个数右边的数这么多经验值，需要注意删除一个数后它左右两边的数就直接相连了，如果一个数左边或者右边没有数，那么从这一边获得的经验值为 0 。

现在给定最原始的排列和小明删除数字的顺序，请你帮小明算一下他最后能获得的总经验值是多少。

【输入格式】

第一行一个整数 $n(1 \leq n \leq 10^5)$

第二行 n 个整数， $p_1, p_2, \dots, p_n$ 表示最原始的排列

第三行 n 个整数， $d_1, d_2, \dots, d_n$ ； d_i 表示小明第 i 次删除数字 d_i

【输出格式】

输出一行，包含一个整数，表示答案

【样例输入】

3

1 2 3

3 1 2

【样例输出】

4

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,a[100005],p[100005],pre[100005],nxt[100005],x,l,r;
4. long long sum;
5. int main(){
6.     cin>>n;
7.     for(int i=1;i<=n;i++){
8.         cin>>x;
9.         a[i]=x;
10.        p[x]=i;
11.        pre[i]=i-1;
12.        nxt[i]=i+1;
13.    }
14.    nxt[0]=1;
15.    pre[n+1]=n;
16.    for(int i=1;i<=n;i++){
17.        cin>>x;
18.        l=pre[p[x]];
19.        r=nxt[p[x]];
20.        sum+=a[l]+a[r];
21.        nxt[l]=r;
22.        pre[r]=l;
23.    }
24.    cout<<sum;
25.    return 0;
26. }
```

练 77.1 队列复原

【题目描述】

小瓜现在让 1 到 n 这 n 个整数排成一列，但是他只告诉你每个整数的后面那个数是什么(最后一个整数的后面那个数是 0)，请你帮忙复原这个队列。

【输入格式】

第一行一个整数 n ($n \leq 100000$)，表示有 n 个整数。接下来 n 行，每行两个数 i, j ，表示排在整数 i 后面的那个数是 j 。

【输出格式】

n 行，每行一个整数，表示完整的队列。

【样例输入】

```
3
1 2
2 3
3 0
```

【样例输出】

```
1
2
3
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,x,y,nn,b[100005],m,a[100005];
4. int main(){
5.     cin>>n;
6.     m=n;
7.     while(m--){
8.         cin>>x>>y;
9.         a[y]=x;
10.        if(y==0) b[n]=x;
11.    }
12.    nn=n;
13.    while(nn--){
14.        b[nn]=a[b[nn+1]];
15.    }
16.    for(int i=1;i<=n;i++) cout<<b[i]<<endl;
17.    return 0;
18.}
```

谢谢！

