

公告

昵称：T\jl  
园龄：2年10个月  
粉丝：8  
关注：0  
+加关注

<	2020年5月						>
日	一	二	三	四	五	六	
26	27	28	29	30	1	2	
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	
31	1	2	3	4	5	6	

搜索

找找看

谷歌搜索

常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

最新随笔

- 1.深信服入职前编码训练21题--02
- 2.深信服入职前编码训练21题--01
- 3. Leetcode本地阅读器开发 - - 01  
界面设计三
- 4. Leetcode本地阅读器开发 - - 01  
界面设计二
- 5. Leetcode本地阅读器开发 - - 01  
界面设计一
- 6. Leetcode本地阅读器开发 - - 00  
总声明
- 7. PAT 乙级 暂停
- 8. PAT-乙级-1066 图像过滤
- 9. PAT-乙级-1065 单身狗
- 10. PAT-乙级-1064 朋友数

我的标签

GUI(3)  
QT(3)  
Leetcode(2)  
qt目录操作与递归检索(1)

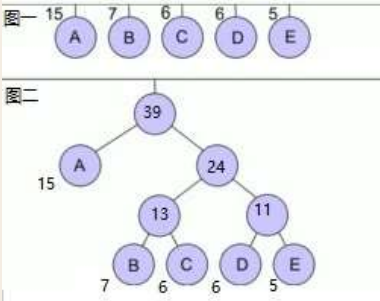
随笔分类

c++面试笔试(5)

深信服入职前编码训练21题--02

题目描述:

有一个节点数组，需要创建一棵最优二叉树，即每个节点的权值乘以节点在树中的长度，然后相加得到的值最小。以下图一为例，节点数组的[A,B,C,D,E]的权值分别为[15,7,6,6,5]，构建好的最优二叉树见下图。



相关代码如下，请补充缺失部分。

```
...
struct node {
    int left, right, parent;
    int val;
};
int build_tree(struct node arr[], int cnt)
{
    while (1) {
        int i;
        int min1 = -1;           //权值最小的节点编号
        int min2 = -1;           //权值第二小的节点编号
        int root_node = 0;       //根节点(没有父节点)的个数

        for (i = 0; i < cnt; ++i) {
            if (arr[i]._____ >= 0)
                continue;
            ++root_node;
            if (min1 < 0) {
                min1 = i;
            } else if (arr[i].val < _____) {
                min2 = min1;
                min1 = i;
            } else if (min2 < 0) {
                min2 = i;
            } else if (arr[i].val < _____) {
                min2 = i;
            }
        }
        if (root_node < _____)
            break;
    }
}
```

c++数据结构(14)  
c++数据结构题(2)  
c++知识点(14)  
HDoj(60)  
LeetCode(7)  
Linux(2)  
ml-链接文件(8)  
PAT-甲级(3)  
PAT-乙级(67)  
python(8)  
Qt(4)  
笔记(10)  
常见算法(8)  
机器学习(6)  
链接文件(5)

随笔档案

2020年5月(2)  
2020年3月(4)  
2018年12月(16)  
2018年11月(9)  
2018年10月(57)  
2018年9月(1)  
2018年3月(4)  
2018年2月(16)  
2017年12月(78)  
2017年11月(7)  
2017年10月(10)  
2017年9月(21)  
2017年7月(1)

相册

小伙子(2)

最新评论

1. Re:智能车起步第一篇---认识K6  
0  
O(∩\_∩)O~  
--Tzv

阅读排行榜

1. 数据结构第六篇——顺序存储结构  
与链式存储结构的特点(5104)  
2. extern 详解 ( 内有extern "C" 讲  
解 ) (4778)  
3. c++重载 ( 以运算符重载为主 )  
(3307)  
4. 线性表——一元多项式的求和(2  
538)  
5. 多继承的构造和析构函数调用顺  
序(2230)

评论排行榜

1. K60(1)

推荐排行榜

1. 数据结构第一篇——线性表的逻辑  
结构(2)  
2. 模拟电子电路 ( 上 ) (1)  
3. 机器学习二 贝叶斯分类器(1)  
4. extern 详解 ( 内有extern "C" 讲  
解 ) (1)

```
arr[cnt].left = min2;
arr[cnt].right = min1;
arr[cnt].val = arr[min1].val + _____;
arr[cnt].parent = -1;
arr[min1].parent = cnt;
arr[min2].parent = cnt;
++cnt;
}
return cnt;
}
...

```



输入描述:

第一行为数据个数 第二行为权值（整数）

输出描述:

构建的二叉树（用于绘图软件生成对应的二叉树图形）

示例1



输入

```
5
15 7 6 6 5
```

输出

```
```mermaid
graph TD
    n0[n0:15]
    n0 --> n8
    n1[n1:7]
    n1 --> n6
    n2[n2:6]
    n2 --> n5
    n3[n3:6]
    n3 --> n6
    n4[n4:5]
    n4 --> n5
    n5((11))
    n5 --> n7
    n6((13))
    n6 --> n7
    n7((24))
    n7 --> n8
    n8((39))
    ...

```

说明

1.grath TD下面的输出都是\t开头  
2.n0 ---> n8 的意思是n0的父节点是n8



分析：这道题是典型的哈夫曼树的构造，只需补充完树的构建部分即可。

解答：



```

1 #include <stdio.h>
2 #include <limits.h>
3 #include <assert.h>
4 #include <malloc.h>
5
6 struct node {
7     int left, right, parent;
8     int val;
9 };
10
11 void tree_print(const struct node arr[], int cnt)
12 {
13     int i;
14     for (i = 0; i < cnt; ++i) {
15         fprintf(stderr, "%d: {left:%d,right:%d,parent:%d,val:%d}\n"
16             , i, arr[i].left, arr[i].right, arr[i].parent, arr[i].val);
17     }
18 }
19
20 void tree_output(FILE *fp, const struct node arr[], int old_cnt, int cnt)
21 {
22     int i;
23     fprintf(fp, "`mermaid\n");
24     fprintf(fp, "graph TD\n");
25     for (i = 0; i < cnt; ++i) {
26         if (i < old_cnt)
27             fprintf(fp, "\tn%d[n%d:%d]\n", i, i, arr[i].val);
28         else
29             fprintf(fp, "\tn%d((%d))\n", i, arr[i].val);
30
31         if (arr[i].parent >= 0) {
32             fprintf(fp, "\tn%d --> n%d\n", i, arr[i].parent);
33         }
34     }
35     fprintf(fp, "`\n");
36 }
37
38 int build_tree(struct node arr[], int cnt);
39
40 static int input(int **arr, int *size)
41 {
42     int i;
43     int ret;
44
45     ret = fscanf(stdin, "%d\n", size);
46     if (ret != 1)
47         return -1;
48     if (*size <= 0)
49         return -1;
50     *arr = (int *)malloc(sizeof(int) * (*size));
51     for (i = 0; i < *size; ++i) {
52         fscanf(stdin, "%d ", &(*arr)[i]);
53     }
54     return 0;
55 }
56
57 int main(int argc, char *argv[])
58 {
59     int *vals = NULL;
60     int cnt = 0;
61     struct node *arr;
62     int i;
63
64     if (input(&vals, &cnt) < 0) {
65         fprintf(stderr, "input error\n");
66         return 0;
67     }
68     arr = (struct node *)malloc(sizeof(struct node) * cnt * 3);
69
70     for (i = 0; i < cnt; ++i) {
71         arr[i].left = -1;
72         arr[i].right = -1;
73         arr[i].parent = -1;

```

```

73     arr[i].val = vals[i];
74 }
75
76 int newcnt = build_tree(arr, cnt);
77 tree_output(stdout, arr, cnt, newcnt);
78 free(vals);
79 free(arr);
80 return 0;
81 }
82
83 // 建树, 返回树的根节点
84 int build_tree(struct node arr[], int cnt)
85 {
86     while (1) {
87         int i;
88         int min1 = -1;           // 权值最小的节点编号
89         int min2 = -1;           // 权值第二小的节点编号
90         int root_node = 0;       // 根节点 (没有父节点) 的个数
91
92         for (i = 0; i < cnt; ++i) {
93             if (arr[i].parent >= 0)    // 拥有父节点, 则跳过这个结点
94                 continue;
95             ++root_node;               // 根节点加1
96             if (min1 < 0) {
97                 min1 = i;
98             } else if (arr[i].val < arr[min1].val) {    // 出现比第一更小的, 更新
99                 min2 = min1;
100                min1 = i;
101            } else if (min2 < 0) {
102                min2 = i;
103            } else if (arr[i].val < arr[min2].val) {    // 当前节点权值比第二小节点更小
104                min2 = i;
105            }
106        }
107        if (root_node < 2)
108            break;
109        arr[cnt].left = min2;
110        arr[cnt].right = min1;
111        arr[cnt].val = arr[min1].val + arr[min2].val;
112        arr[cnt].parent = -1;
113        arr[min1].parent = cnt;
114        arr[min2].parent = cnt;
115        ++cnt;
116    }
117    return cnt;
118 }

```

[好文要顶](#)
[关注我](#)
[收藏该文](#)


T\jl  
[关注 - 0](#)  
[粉丝 - 8](#)

[+加关注](#)

0

[推荐](#)

0

[反对](#)

« 上一篇: [深信服入职前编码训练21题-01](#)

posted @ 2020-05-13 17:06 T\jl 阅读(8) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

[注册用户登录后才能发表评论, 请\[登录\]\(#\)或\[注册\]\(#\), \[访问\]\(#\) \[网站首页\]\(#\)。](#)

【推荐】AI大咖带您解锁深度学习算法, 华为开发者学院直播间福利仅29元!

【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】斩获阿里offer的必看12篇面试合辑

【推荐】开放下载!《阿里巴巴大数据及AI实战》深度解析典型场景实践



#### 相关博文：

- Qunar入职前自学笔记
- 入职前要做的几件事
- 实习及入职前的高效习惯
- 入职前你可能需要准备这些.....
- 入职前做好充分准备（转）
- » 更多推荐...

精品问答：前端开发必懂之 HTML 技术五十问



#### 最新 IT 新闻：

- 陆正耀剪了个指甲 钱治亚成了瑞幸造假的弃子
- 启动B计划 京东要找新动能？
- 瑞幸开卖面膜 承认造假前曾大幅扩充经营范围
- 被疫情“美化”的腾讯20Q1业绩：能有持续性吗？
- 币圈大军涌入口罩圈：炒币亏了200万，炒口罩赚了6000万
- » 更多新闻...

#### 历史上的今天：

2020-05-13 深信服入职前编码训练21题--01