

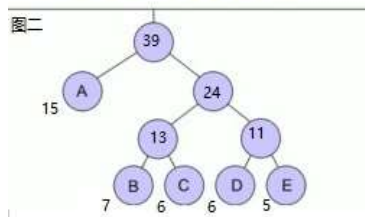
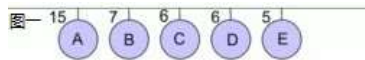


3/3 [编程题]最优二叉树

时间限制：C/C++ 1秒，其他语言2秒

空间限制：C/C++ 32M，其他语言64M

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图二。



相关框架代码已经给出，请补充缺失部分，保证程序正常运行，输出预期结果。

...

```
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;
        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;
    }
}
```





...

#### 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）

#### 输入例子1:

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

#### 输出例子1:

```
```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))
```
```

#### 例子说明1:

- 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



C++11(clang++ 3.9) ▾



重置



自测



```
1 #include <stdio.h>
2 #include <limits.h>
3 #include <assert.h>
4 #include <malloc.h>
5
```



```
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, "```\nmermaid\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     if (input(&vals, &cnt) < 0) {
```





```
66     }
67     arr = (struct node *) malloc(sizeof(struct node) * cnt * 3);
68
69     for (i = 0; i < cnt; ++i) {
70         arr[i].left = -1;
71         arr[i].right = -1;
72         arr[i].parent = -1;
73         arr[i].val = vals[i];
74     }
75
76     build_tree(arr, cnt);
77
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)
94                 continue;
95             ++root_node;
96             if (min1 < 0) {
97                 min1 = i;
98             } else if (arr[i].val < arr[min1].val) {
99                 min2 = min1;
100                min1 = i;
101            } else if (min2 < 0) {
102                min2 = i;
103            } else if (arr[i].val < arr[min2].val) {
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 }
```



提交运行

交卷



牛客网  
NOWCODER

[首页](#)[题库](#)[面试](#)[学习](#)[求职](#)[讨论区](#)[发现](#)[帖子、文章、试题、公司、牛友](#)[资源导航](#)

收起答题卡 ▾

编程题3道

1

2

3



求职之前，先上牛客



扫一扫，把题目装进口袋

[关于我们](#) [加入我们](#) [意见反馈](#) [企业服务](#) [联系我们](#) [免责声明](#)  
[友情链接](#)

公司地址：北京市朝阳区大屯路东金泉时代广场3单元北京牛客科技有限公司

联系方式：010-60728802(电话) [admin@nowcoder.com](mailto:admin@nowcoder.com)

牛客科技©2020 All rights reserved

京ICP备14055008号-4 增值电信业务经营许可证

京公网安备 11010502036488号

