

# 实验课4

## 二叉树、树、DP、贪心

数据结构与算法 2023年秋

# Outline

参考难度:

- 简单
- 中等
- 困难

- |                     |                 |       |
|---------------------|-----------------|-------|
| • Task 1. 二叉树的深度    | depth.cpp       | (有模板) |
| • Task 2. 排序二叉树     | bst.cpp         | (选做题) |
| • Task 3. 多叉树转二叉树遍历 | c2t.cpp         | (有模板) |
| • Task 4. 最长递增子序列   | longest.cpp     | (有模板) |
| • Task 5. 任务安排问题    | taskprocess.cpp | (有模板) |

# 提交测试注意

- 测试及反馈结果时间点：

**16:00 / 16:30 / 17:00  
/ 17:30 / 18:00**

- 后提交代码会覆盖前面代码
- 测试时以收集到的代码为准

# 评分规则

- 课上**当堂**提交，**正常**评分（当堂提交以**最终**提交版本为准）
- 课后**当天晚12点前**交，评分结果 **x 0.8**
- 课后**当周周日晚12点前**交，评分结果 **x 0.6**
- **其他时段提交不评分**
- **注意：评分以实验室提交作业为准（在线测评只提供个人代码调试），学号命名文件夹，文件夹包含每题cpp文件**

# 自我提交测试

- 同学们课上自行使用在线平台**测试**，下课（或完成后）将最终代码于实验室**提交**（依据最终提交给分）

在线平台使用链接：<https://matrix.sysu.edu.cn/>

数据结构与算法实验（薛磊）

课程信息 课程题目 抄袭检查 数据导出 IP管理

下载 改期 重评

题目名称	题目类型	开始时间	截止时间	提交进度	操作
多叉树转二叉树遍历-c2t.cpp	实时编程题	11-22 15:11	11-29 15:11	0%	<a href="#">查看提交</a> <a href="#">修改发布</a> <a href="#">取消发布</a>
二叉树的深度-depth.cpp	实时编程题	11-22 15:11	11-29 15:11	0%	<a href="#">查看提交</a> <a href="#">修改发布</a> <a href="#">取消发布</a>
排序二叉树-bst.cpp	实时编程题	11-22 15:11	11-29 15:11	0%	<a href="#">查看提交</a> <a href="#">修改发布</a> <a href="#">取消发布</a>
任务安排问题-taskprocess.cpp	实时编程题	11-22 15:11	11-29 15:11	0%	<a href="#">查看提交</a> <a href="#">修改发布</a> <a href="#">取消发布</a>
最长递增子序列-longest.cpp	实时编程题	11-22 15:11	11-29 15:11	0%	<a href="#">查看提交</a> <a href="#">修改发布</a> <a href="#">取消发布</a>

**注意！username**请使用学号（只用学号）

# 注意事项!!!

- 不得使用c++/STL 库（会有检测程序）
  - 允许的5个库白名单: `stdlib stdio iostream memory malloc`
  - 如`string cmath bits/stdc++`等都是不可用
- 现已可以正常使用gets和puts函数

# Task 1 二叉树的深度

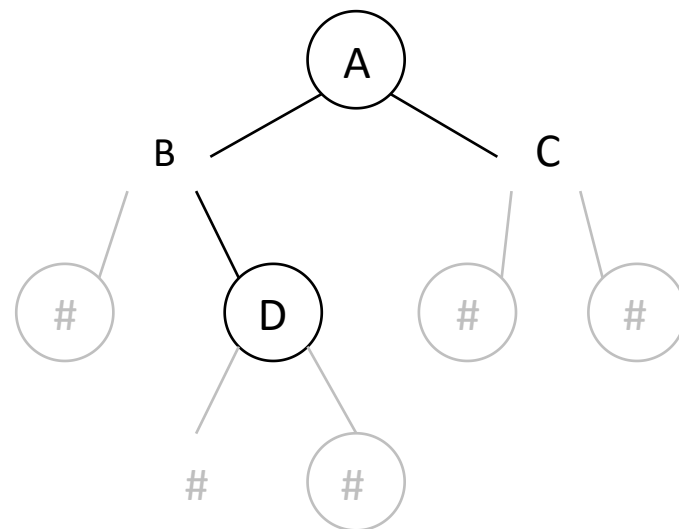
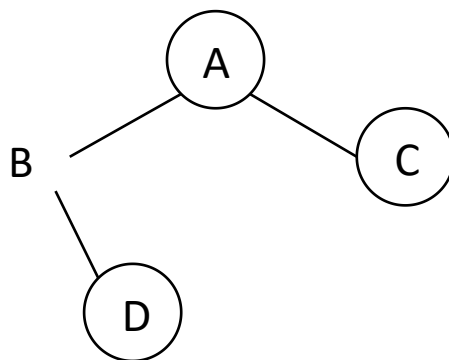
depth.cpp

- **输入格式：** 第一行一串字符
  - 这串字符表示扩展二叉树的前序序列，其中空结点用#表示
- **输出格式：**
  - 一个整数代表二叉树的深度
- **样例输入：**

AB#D##C##

(长度 $\leq 200$ )

- **样例输出：**  
3



# Task 2 排序二叉树 -- 选做题

bst.cpp

- 输入格式：
  - 第一行一个整数 $n$  ( $1 \leq n \leq 1000$ )
  - 第二行包含用空格分开的 $n$ 个整数 (范围在  $[0, 10^5]$ ) (各不相同)
- 输出格式：
  - 第一行用空格分开的 $n$ 个数, 表示生成的二叉排序树的中序遍历
  - 第二行输出层次信息。第三行输出度数信息。

- 样例输入：

```
5
4 1 8000 2000 7
```

- 样例输出：

```
1 4 7 2000 8000
2 1 4 3 2
0 2 0 1 1
```

第二、三行输出顺序随第一行而不是输入  
例如, 8000节点位于第一行第5位, 故其对应的层次2、度数1位于第二、三行的第5位



# Task 3 多叉树转二叉树遍历

c2t.cpp

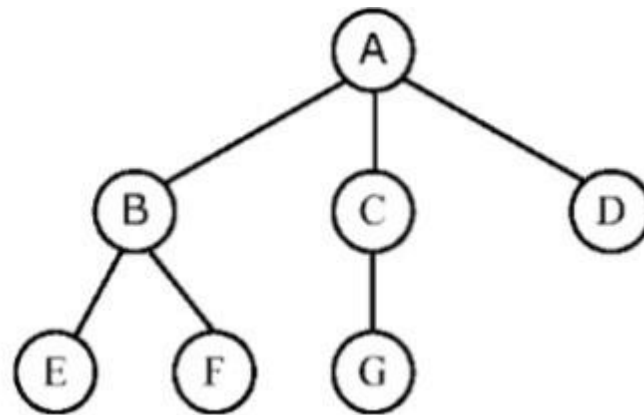
- **输入格式：** 按顺序输入树的父亲表示
  - 第一行为节点个数（100以内并且大于0），之后一个节点一行，第1个字符为数据，接下来是一个空格，接下来是父亲的编号，根节点的父亲编号为0。
  - 保证节点i的父亲编号  $< i$ 。
- **输出格式：**
  - 转化为二叉树后，分别输出二叉树先序、中序、后序遍历结果（3行输出）

• **样例输入：**

```
7
A 0
B 1
C 1
D 1
E 2
F 2
G 3
```

**样例输出：**

```
ABEFCGD
EFGCDA
FEGDCBA
```



# Task 4 最长递增子序列

longest.cpp

- 输入格式： 第一行一个正整数 $n$ 。
  - 第二行包含用空格分开的 $n$ 个整数  $a_1, a_2, \dots, a_n$  (注意可能有重复元素)
- 输出格式：
  - 输出一个整数，表示序列 $a_1, a_2, \dots, a_n$ 的最长递增子序列的长度。
- 数据范围：  $1 \leq n \leq 12000$ 。  $-10^9 \leq a_1, a_2, \dots, a_n \leq 10^9$ 。

• 样例输入：

7

3 1 5 2 6 7 4

• 样例输出：

4

要求严格递增，即子序列中不能有相等元素

## Task 4 测试样例

输入:

8

14 2 -3 12 11 13 11 19

输出:

4

输入:

15

31 -9 -16 27 5 -37 17 42 -18 32 40 -16 1 -4 43

输出:

6

输入:

25

10 -40 40 -35 51 -14 40 -45 62 10 -61 71 22 -20 9 -1 37 13 8 62 -6 -37 -27 57 -70

输出:

7

# Task 5 任务安排问题

taskprocess.cpp

- 有 $n$  个任务，编号为 $1\sim n$ ，需要被完成。假设：
  - 任务 $i$  只能在时刻  $r_i$  后开始处理。任务 $i$  需要  $p_i$  个时间单位才能完成。
  - 同一个时间单位只能处理一项任务；一项任务可分多次处理（可暂停，之后继续处理）
- 设计方案使得 $\sum t_i$  最小， $t_i$  为任务 $i$  被完成的时刻。
- 输入格式：
  - $n$
  - $r_1\ r_2\ \dots\ r_n$
  - $p_1\ p_2\ \dots\ p_n$
- 输出格式：一个数，表示最小的 $\sum t_i$ 。
- 数据范围：
  - $1 \leq n \leq 100$
  - $0 \leq r_i \leq 200.$      $1 \leq p_i \leq 200$

样例输入：

3  
1 2 3  
4 2 2

样例输出：

19

# Task 5 测试样例

输入:

5

5 8 8 6 0

8 6 6 1 3

输出:

70

输入:

10

15 2 8 8 20 3 20 14 15 1

17 12 16 7 18 4 2 1 6 19

输出:

423

输入:

15

28 1 3 18 12 2 20 0 25 30 21 27 13 0 8

5 3 2 19 27 10 5 10 6 29 10 30 7 22 5

输出:

961