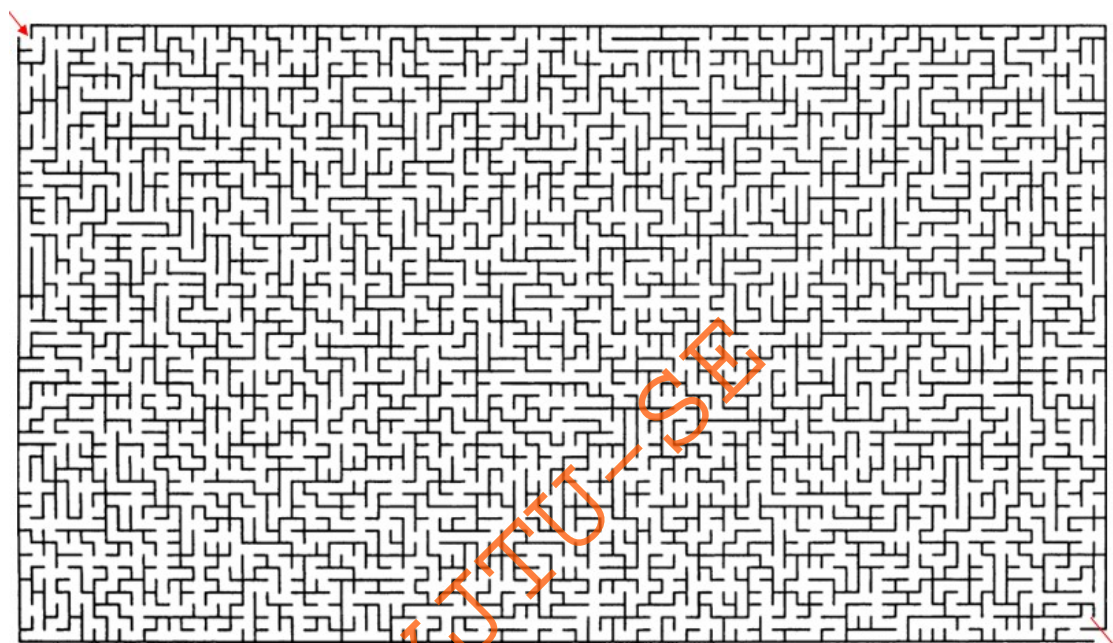


2015 级软件工程专业数据结构与算法课程综合训练

第一部分：实验题目

一. 题目：迷宫

如下图所示即为一个迷宫，题目要求使用 Union-Find 数据结构完成一个迷宫的生成。如图
所示迷宫的入口点位于左上角，出口点是在图的右下角，该图所示的迷宫是由 50×88 个单
元组成的矩形，在该矩形中，左上角的单元被连通到右下角的单元，而且这些单元与相邻的
单元通过墙壁分离开来。



下面将通过输入、输出等方面描述迷宫题目的要求。

1. 输入：

输入两个整数，代表所生成迷宫对应矩阵的大小

2. 输出：

- 1) 用合理的方式展现生成的迷宫；
- 2) 给出迷宫可以走通的路径。以 SEN...（代表向南，然后向东，然后再向北，等等）的形式给出输出结果。

3. 题目中所使用数据结构的建议：

生成迷宫的算法辅助数据结构采用 Union-Find ADT，该抽象数据类型的解释在课本的
第 6 章，同时，也附上一个 Union-Find 的参考资料，供参考学习。

迷宫自身的存储以及路径生成所需的数据结构请自行思考。

二. 题目：微型编程语言解释器

解释器（英语：Interpreter），又译为直译器，是一种电脑程序，能够把高级编程语言
一行一行直接转译运行。解释器不会一次把整个程序转译出来，只像一位“中间人”，每次
运行程序时都要先转成另一种语言再作运行，因此解释器的程序运行速度比较缓慢。它每转

译一程序叙述就立刻运行，然后再转译下一行，再运行，如此不停地进行下去。该题目要求实现的是一个能够解释执行具有赋值语句、函数定义语句以及函数执行语句的小型解释器。下面将通过解释器的输入、输出以及限制等方面描述该问题。

1.输入：

输入是由一组赋值语句、函数定义语句以及函数调用语句构成。

赋值语句具有如下的定义形式：ASSIGN <variable> <expression>。variable 是用一个仅包含英文字母所构成的长度不超过 8 个字符的字符串构成；expression 是一个算术四则混合运算表达式，其包含的运算符就是+、-、*、/ 四个算术运算法，其操作数要么是正数（只需要考虑整数形式）要么是已定义的 variable。表达式中的运算符和操作数使用空格的形式进行分割。一个赋值语句只占用一行，具体举例如下：

```
ASSIGN X 1
ASSIGN X X + 2
ASSIGN Number (X + 2) * (X - 2)
```

函数调用语句的定义形式：CALL <function>。function 即为函数名字，函数名字和 variable 名字的定义要求是一样的。函数调用语句也只占用一行，具体举例如下：

```
CALL TryThis
```

函数定义语句是由一组语句组成，具体要求如下：

- 1) 第一行必须是 DEFINE <function>
 - 2) 最后一行必须是 END
 - 3) 除第一行和最后一行之外的所有行只能是赋值语句或者函数调用语句
- 具体举例如下：

```
DEFINE IncrementX
ASSIGN X X+1
END

DEFINE FF
CALL IncrementX
ASSIGN Y X*X
END
```

备注：输入的源代码中可以包含空行，但这些空行在解释执行过程中都应该被忽略。

2.输出：

解释器执行后，其输出内容应该就是对每一条语句执行结果的显示。具体解释如下：

- 1) 每当执行一条 ASSIGN 语句，解释器都应该输出如下格式的内容：“Assigning <value> to <variable>”
- 2) 每当执行到函数定义语句，解释器都应该输出如下格式的内容：“Defining function <function name>”
- 3) 每当执行到函数调用语句，解释器都应该输出如下格式的内容：“Calling function <function name>”

假设一个源代码如下所示：

```
ASSIGN X 1
ASSIGN Y 1

DEFINE Fib
ASSIGN TMP Y
ASSIGN Y X+Y
```

```

ASSIGN X TMP
END
□
CALL Fib
ASSIGN W X
CALL Fib
ASSIGN Z W * Y - X * X

```

在上面的输入下，解释器的执行结果应该如下：

```

Assigning 1 to X
Assigning 1 to Y
Defining Fib
Calling Fib
Assigning 1 to TMP
Assigning 2 to Y
Assigning 1 to X
Assigning 1 to W
Calling Fib
Assigning 2 to TMP
Assigning 3 to Y
Assigning 2 to X
Assigning -1 to Z

```

3.解释器可以忽略如下检查：

- 1) 输入的源代码格式总是正确的；
- 2) 每个变量在访问之前总是有值的；
- 3) 每个函数在输入的源代码中只会被定义一次；
- 4) 每个函数在调用之前总是被定义好的。
- 5) 不支持递归函数的调用，不管是直接的递归调用还是间接的递归调用。

4.可以提高分数的扩展要求：

使得该编程语言支持循环语句，具体的循环语句格式为：FOR <expression> <statement> 循环语句将根据expression计算的结果值执行多次statement语句。其中，statement要么是一条赋值语句，要么是一个函数调用语句；expression只有在进入循环时才会被计算，而且仅被计算这一次。举例如下：

```

FOR 10 ASSIGN N N + N
FOR / N 2 ASSIGN X X + 1
FOR 8 CALL Fib

```

当执行循环语句时，输出的要求只具体到输出循环语句中对赋值语句或者函数调用语句的输出，而不用专门写对循环语句的输出内容。

5.题目中所使用数据结构的建议：

变量名和函数名的存储可以借助散列表的方式完成，称其为符号表。其他信息的组织方式请自行思考。

第二部分 实验报告

一 封面内容：

标题：数据结构课程综合训练

组名：××××

成员：（包括学号、姓名和班级）

实验报告提交日期：年/月/日

联系电话：××××

二 实验报告内容：

（一）正文内容：

1. 实验名称：（即题目名称）

2. 需求和规格说明

描述问题，简述题目要解决的问题是什么。规定软件做什么。原题条件不足时补全。

3. 设计

3.1 设计思想：存储结构、主要算法基本思想；

3.2 设计表示：每个类型的规格说明；

3.3 实现注释：各项功能的实现程度、在完成基本要求的基础上还实现了什么功能；

3.4 详细设计表示：主要算法的框架；

4. 调试报告：调试过程中遇到的主要问题是如何解决的；对于设计和编码的回顾讨论和分析；时空分析；改进设想；

5. 运行结果展示：通过截图的方式全方位展示程序的运行结果。

（二）实验总结：通过这次实验，你有哪些收获？对这次实验题目有什么意见和建议？

（三）参考文献：

三 实验报告提交形式：

实验报告要求认真撰写并提交实验报告打印版和电子版；其中打印版开学统一提交，电子版则随时可以发送到提交作业的邮箱中。

四 试验报告提交时间：

2017 年 3 月 1 日之前。

第三部分 实验形式

一 自由组合，每组人数控制在 2-3 人

二 要求完成每个实验题目，希望组内之间互相讨论

三 每组只需提交一份实验报告

第四部分 成绩评定

一 成绩=验收+报告

验收：以小组形式，进行现场验收，按老师要求运行源程序并解释老师的提问。