

软件工程导论 作业 4

PB18151866 龚小航

题 1:

某计算机语言中规定，“标识符是由字母开头，后跟字母或数字的任意组合构成。有效字符数为 8 个，最大字符数为 80 个。不能是保留字”。请用等价类划分方法对标识符命名是否正确进行测试。要求给出等价类表，和具体的覆盖数据。

输入条件	有效等价类	标号	无效等价类	标号
标识符 一串 string	字母开头	1	非字母开头	5
			空标识符	6
	只包含字母或数字	2	包含其他特殊字符	7
			包含非打印字符	8
			包含中文字符	9
	有效字符数 8~80 个	3	有效字符数<8	10
			有效字符数>80	11
	标识符整体不是保留数	4	标识符整体是保留字	12

覆盖有效等价类的测试数据:

输入数据	覆盖等价类	预期输出
ValidIdentifier1	1 2 3 4	合法

覆盖无效等价类的测试数据:

输入数据	覆盖等价类	预期输出
1IllegalIdentifier	2 3 4 5	非法
(空)	2 4 6	非法
a!@123aaa	1 3 4 7	非法
abcde□12345	1 3 4 8	非法
a 中文字符 12345	1 3 4 9	非法
a12345678	1 2 4 10	非法
abcdefghijklmnopqrstuvwxyz1 abcdefghijklmnopqrstuvwxyz2 abcdefghijklmnopqrstuvwxyz3	1 2 4 11	非法
register	1 2 3 12	非法

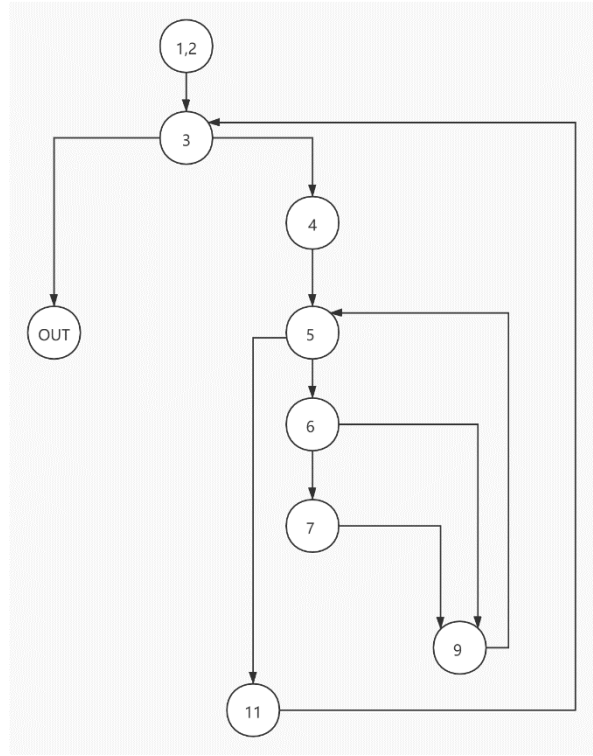
题 2:

给出以下两个代码的**环路复杂度**，并给出所有的**独立路径**，同时对于每条独立路径给出**完整测试用例**以及**对应输出**。

代码一：

```
1 int i = 0;
2 int n = 4;
3 while (i < n-1) {
4     j = i + 1;
5     while(j < n) {
6         if (A[i] < A[j]) {
7             swap(A[i],A[j]);
8         }
9         j = j + 1
10    }
11    i = i + 1;
12}
```

按行数表示的流图如下所示：



$V(G) = E - N + 2 = 11 - 9 + 2 = 4$

给出一组 4 条独立路径：

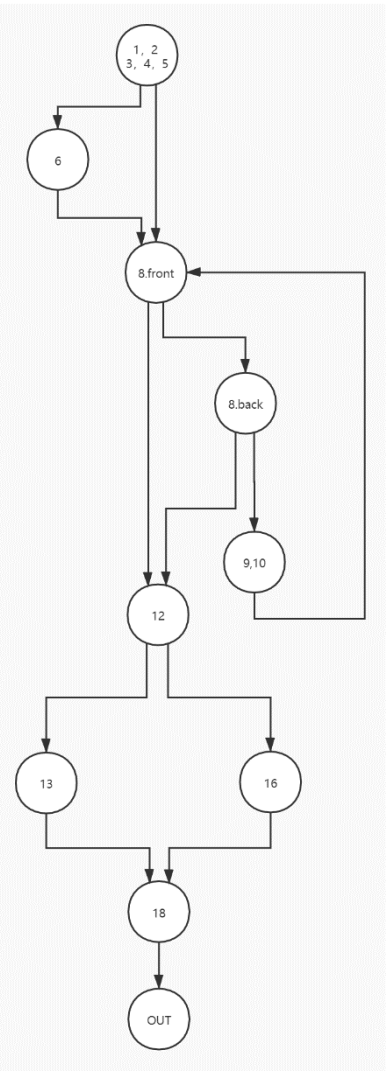
- 路径 1: 1 → 2 → 3 → OUT
- 路径 2: 1 → 2 → 3 → 4 → 5 → 11 → 3 → OUT
- 路径 3: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 9 → 5 → 9 → 11 → 3 → OUT
- 路径 4: 1 → 2 → 3 → 4 → 5 → 6 → 9 → 5 → 6 → 9 → 5 → 6 → 9 → 5 → 11 → 3 → 4 → 5 → 6 → 9 → 5 → 6 → 9 → 5 → 11 → 3 → 4 → 5 → 6 → 7 → 9 → 5 → 11 → 3 → OUT

由于路径 4 包含了所有的分支（实际上就是测试用例 $A = \{4,3,1,2\}$ 的执行路径），即流图上所有边和点都包括，因此这条路径包含了前面的三条路径。因此实际上只需要这一条独立路径。设计测试样例输入 $A = \{4,3,1,2\}$ ，所有分支都被执行。若测试通过，期望的输出为 $A = \{4,3,2,1\}$

代码二：

```
1 public int sum(int n, int upperbound) {
2     int result, i;
3     result = 0;
4     i = 0;
5     if (n < 0) {
6         n = -n;
7     }
8     while(i<n && result <= upperbound) {
9         i = i + 1;
10        result = result + i;
11    }
12    if(result <= upperbound) {
13        System.out.println("The sum is " + result);
14    }
15    else {
16        System.out.println("The sum is too large!");
17    }
18    return result;
19}
```

按行数表示的流图如下所示：由于 while 判断存在局部截断特性，因此将第八行拆为前后两部分。



$V(G) = E - N + 2 = 13 - 10 + 2 = 5$

给出一组 2 条独立路径：

- 路径 1: 1~5 → 8. *front* → 12 → 13 → 18 → OUT
- 路径 2: 1~5 → 6 → 8. *front* → 8. *back* → 9 → 10 → 8. *front* → 8. *back* → 12 → 16 → 18 → OUT

每条测试路径的测试用例以及输出如下表所示：

路径	n	$upperbound$	期望输出（打印前台与返回值）
1	0	100	The sum is 0 $result = 0$
2	-1	0	The sum is too large! $result = 1$