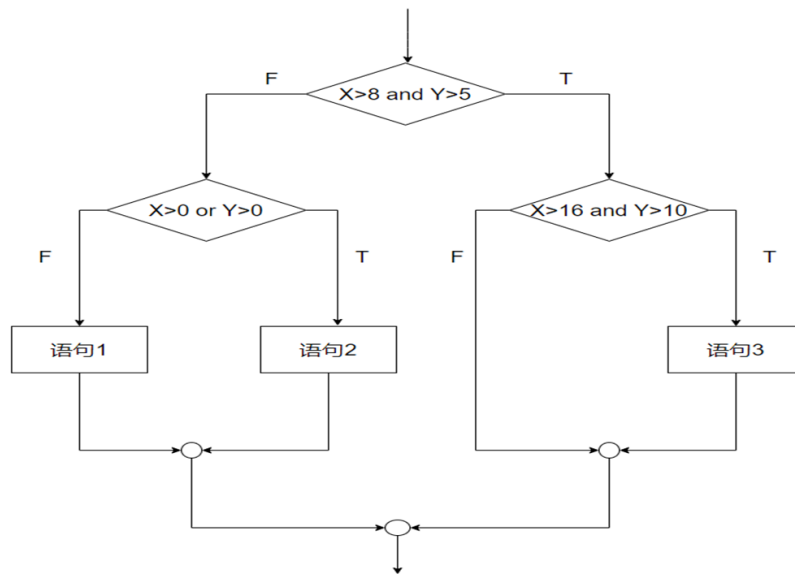


一、



1.语句覆盖:

- ①X=20,Y=11
- ②X=1,Y=-1
- ③X=-1,Y=-1

2.判定覆盖:

- ①X=20,Y=11
- ②X=10,Y=10
- ③X=1,Y=-1
- ④X=-1,Y=-1

3.条件覆盖:

- ①X=20,Y=11
- ②X=20,Y=10
- ③X=16,Y=10
- ④X=10,Y=1
- ⑤X=-1,Y=1
- ⑥X=-1,Y=-1

4.判定/条件覆盖:

- ①X=20,Y=11
- ②X=20,Y=10
- ③X=16,Y=10
- ④X=10,Y=1
- ⑤X=-1,Y=1
- ⑥X=-1,Y=-1

5.条件组合覆盖:

- ①X=20,Y=11
- ②X=20,Y=10
- ③X=10,Y=10
- ④X=10,Y=11
- ⑤X=10,Y=-1
- ⑥X=-1,Y=6
- ⑦X=-1,Y=-1
- ⑧X=1,Y=1

二、

```
int LIS (int [ ] array)
{
1   int result = 0;
2   int [ ] LIS = new int[array.Length];
    //array.Length为求数组长度的操作
3   int i = 0;
4   while (i < array.Length) {
5       LIS[i] = 1;
6       int j = 0;
7       while (j < i) {
8           if (array[i] >= array[j] && LIS[j]+1 > LIS[i])
9               LIS[i] = LIS[j] + 1;
10          j++;
11      }
12      i++;
13  }
14  return max(LIS); //max函数求得LIS中最大
    //元素，可看作基本语句，
    //无需考虑内部实现
}
```

1. 绘制程序流程图

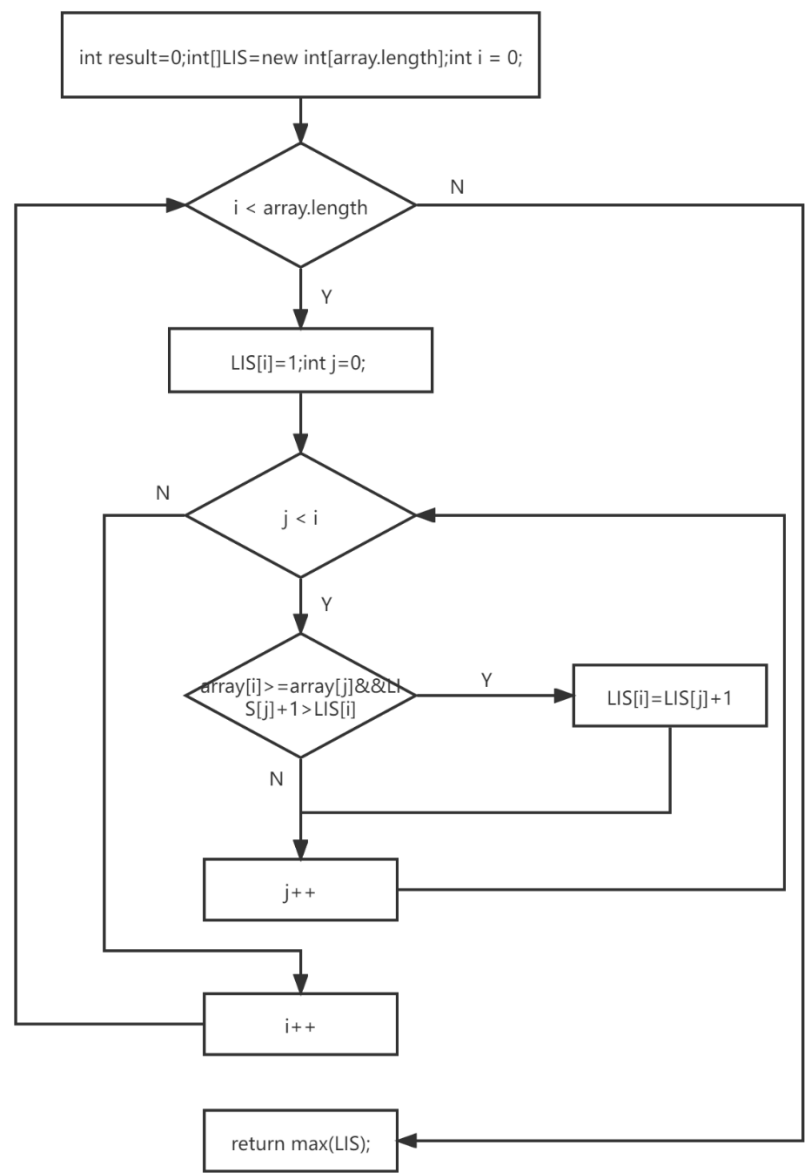


图 1 程序流程图

2.绘制控制流图

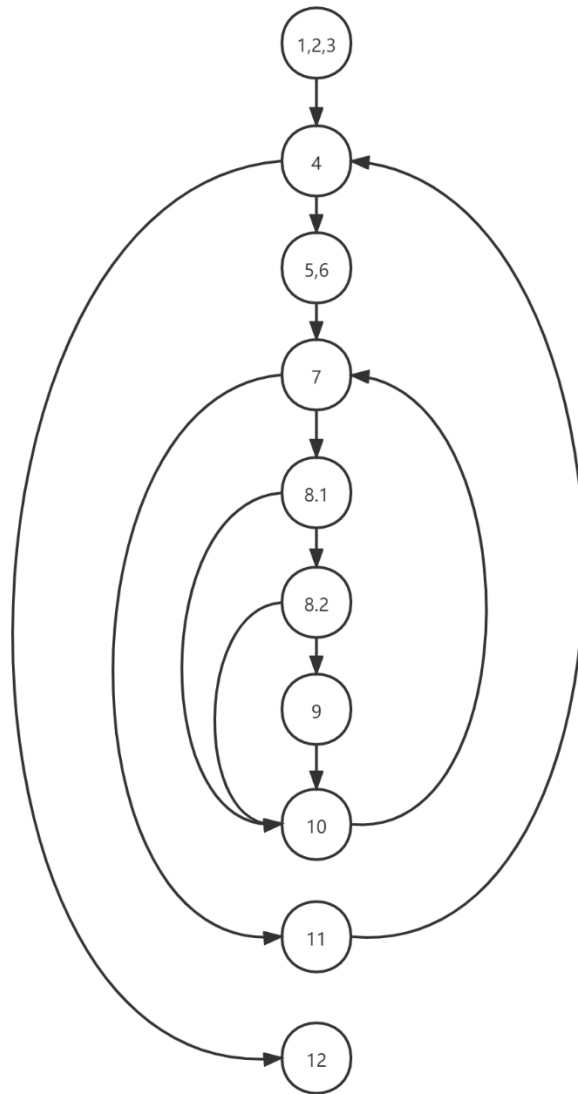


图 2 控制流图

其中 8.1 为程序流程图中编号为 8 的判断语句的前半句，即  $\text{array}[i] \geq \text{array}[j]$

8.2 为后半句，即  $\text{LIS}[j]+1 > \text{LIS}[i]$

### 3. 计算圈复杂度

图中判定节点个数为 4，故圈复杂度为  $4+1=5$

### 4. 列出线性独立路径

- ① 1-2-3-4-12
- ② 1-2-3-4-5-6-7-11-4-12
- ③ 1-2-3-4-5-6-7-8.1-10-7-11-4-12
- ④ 1-2-3-4-5-6-7-8.1-8.2-10-7-11-4-12
- ⑤ 1-2-3-4-5-6-7-8.1-8.2-9-10-7-11-4-12