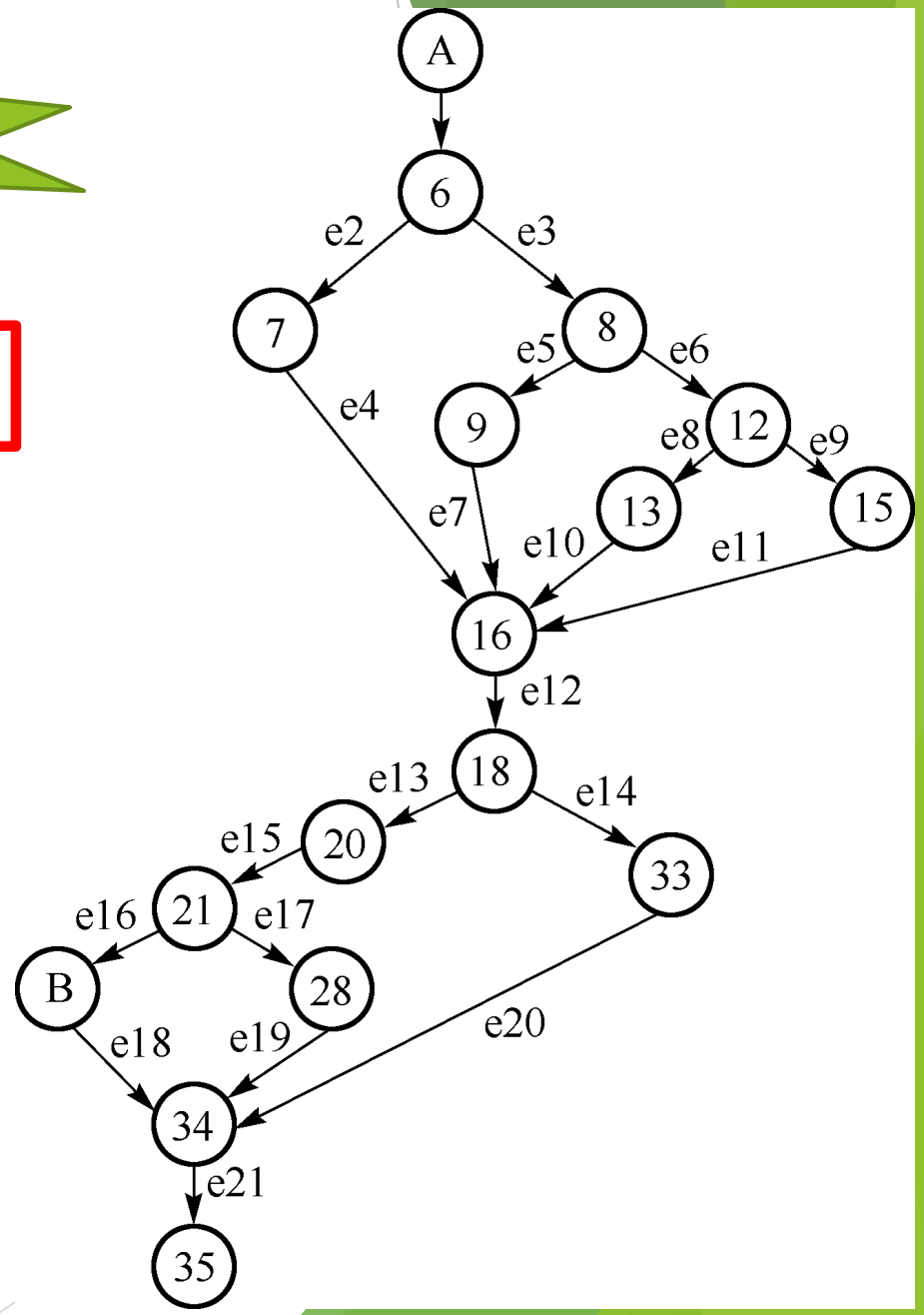
路径可达问题

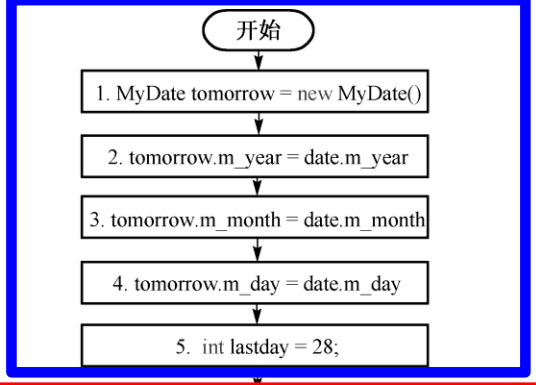
独立路径测试

不可行？

- P1：A, 6, 8, 12, 13, 16, 18, 20, 21, B, 34, 35；
- P2：A, 6, 7, 16, 18, 20, 21, B, 34, 35；
- P3：A, 6, 8, 9, 16, 18, 20, 21, B, 34, 35；
- P4：A, 6, 8, 12, 15, 16, 18, 20, 21, B, 34, 35；
- P5：A, 6, 8, 12, 13, 16, 18, 33, 34, 35；
- P6：A, 6, 8, 12, 13, 16, 18, 20, 21, 28, 34, 35

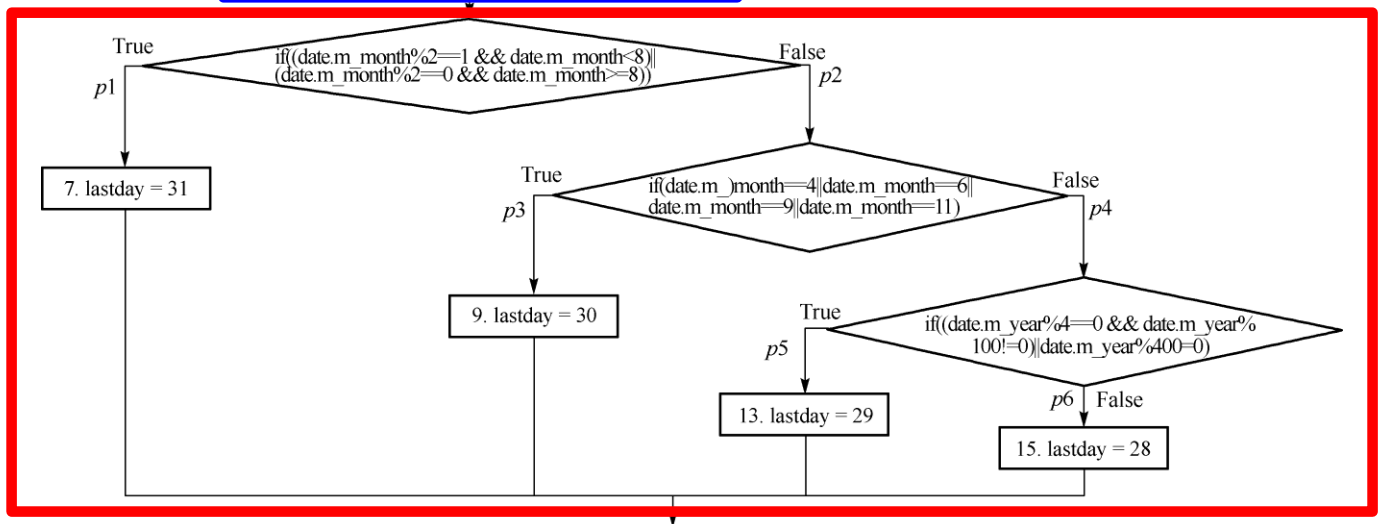
完整路径：12条 ⟸ 独立路径：6条

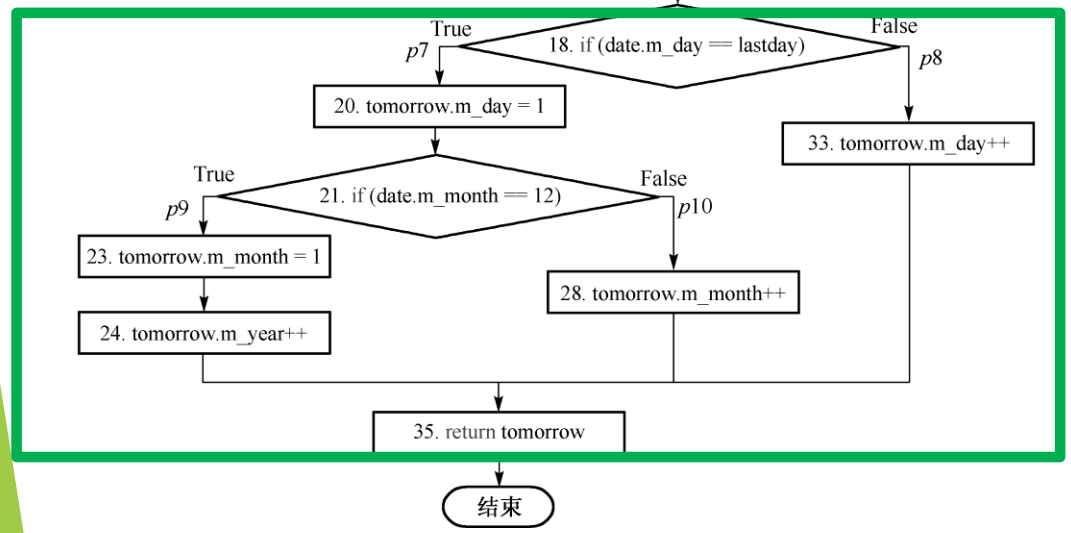




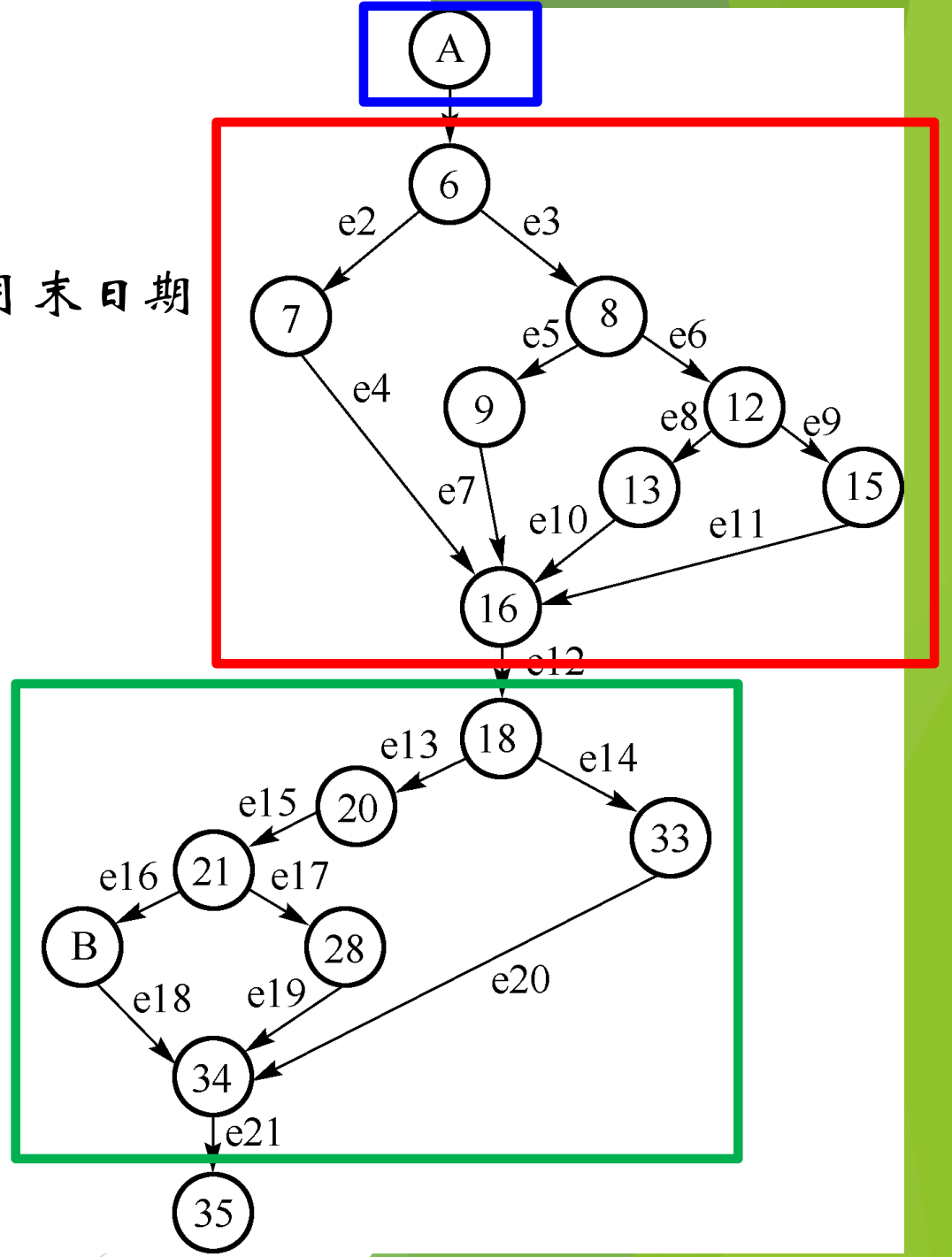
1.基本变量初始化

2.确定输入月份的月末日期



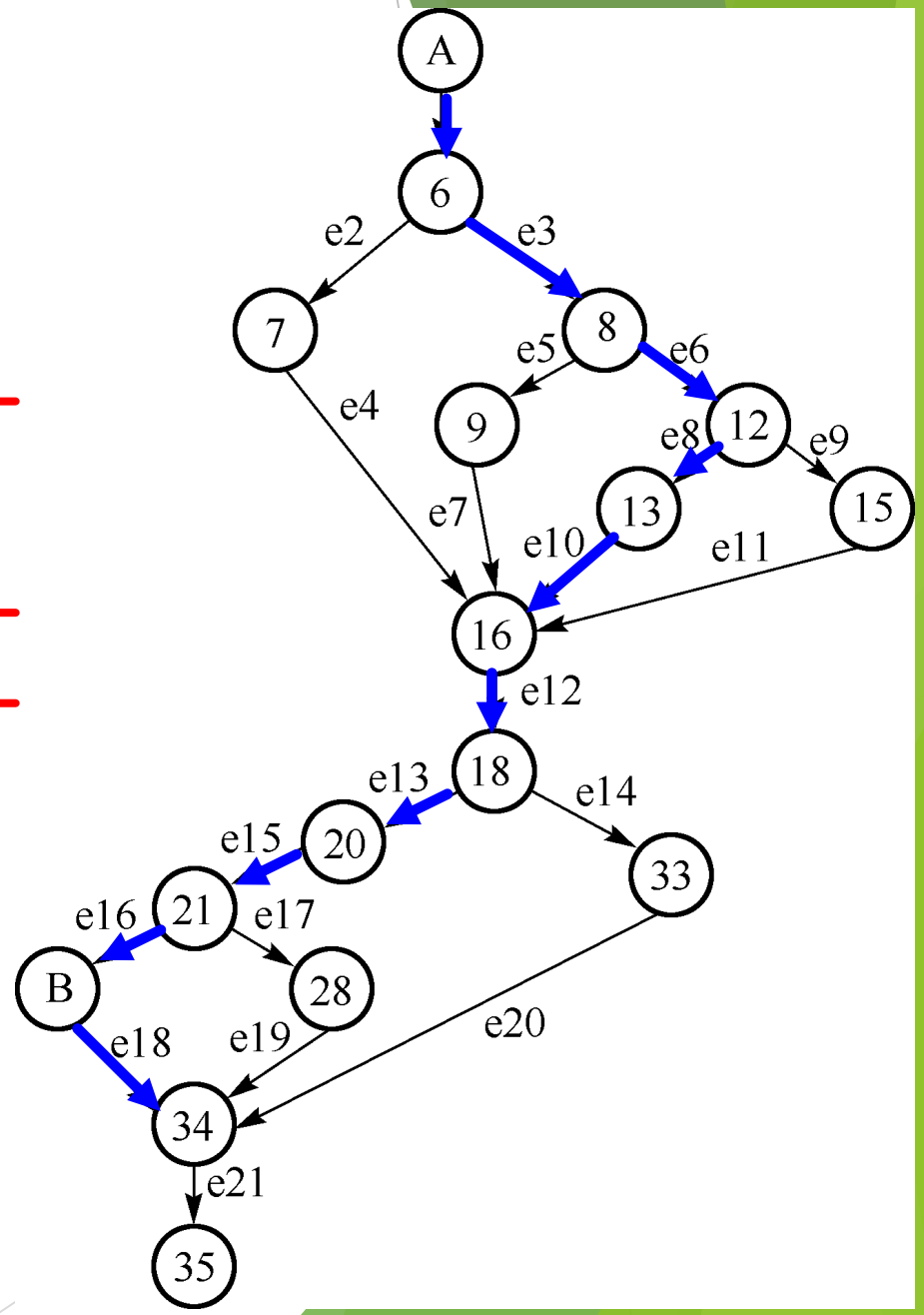


3.判断日期类型并对应处理



局限性：不可行路径问题

- ~~P1：A, 6, 8, 12, 13, 16, 18, 20, 21, B, 34, 35；~~
- P2：A, 6, 7, 16, 18, 20, 21, B, 34, 35；
- ~~P3：A, 6, 8, 9, 16, 18, 20, 21, B, 34, 35；~~
- ~~P4：A, 6, 8, 12, 15, 16, 18, 20, 21, B, 34, 35；~~
- P5：A, 6, 8, 12, 13, 16, 18, 33, 34, 35；
- P6：A, 6, 8, 12, 13, 16, 18, 20, 21, 28, 34, 35



不可行路径问题

- 不可行路径对测试造成什么影响？
- 如何处理不可行路径？
- 不可行路径如何产生的？
- 开发过程中如何避免不可行路径？

1. 不可行路径对测试带来的影响

- 破坏了独立路径测试的完备性和无冗余性
- 增大了测试用例设计的难度

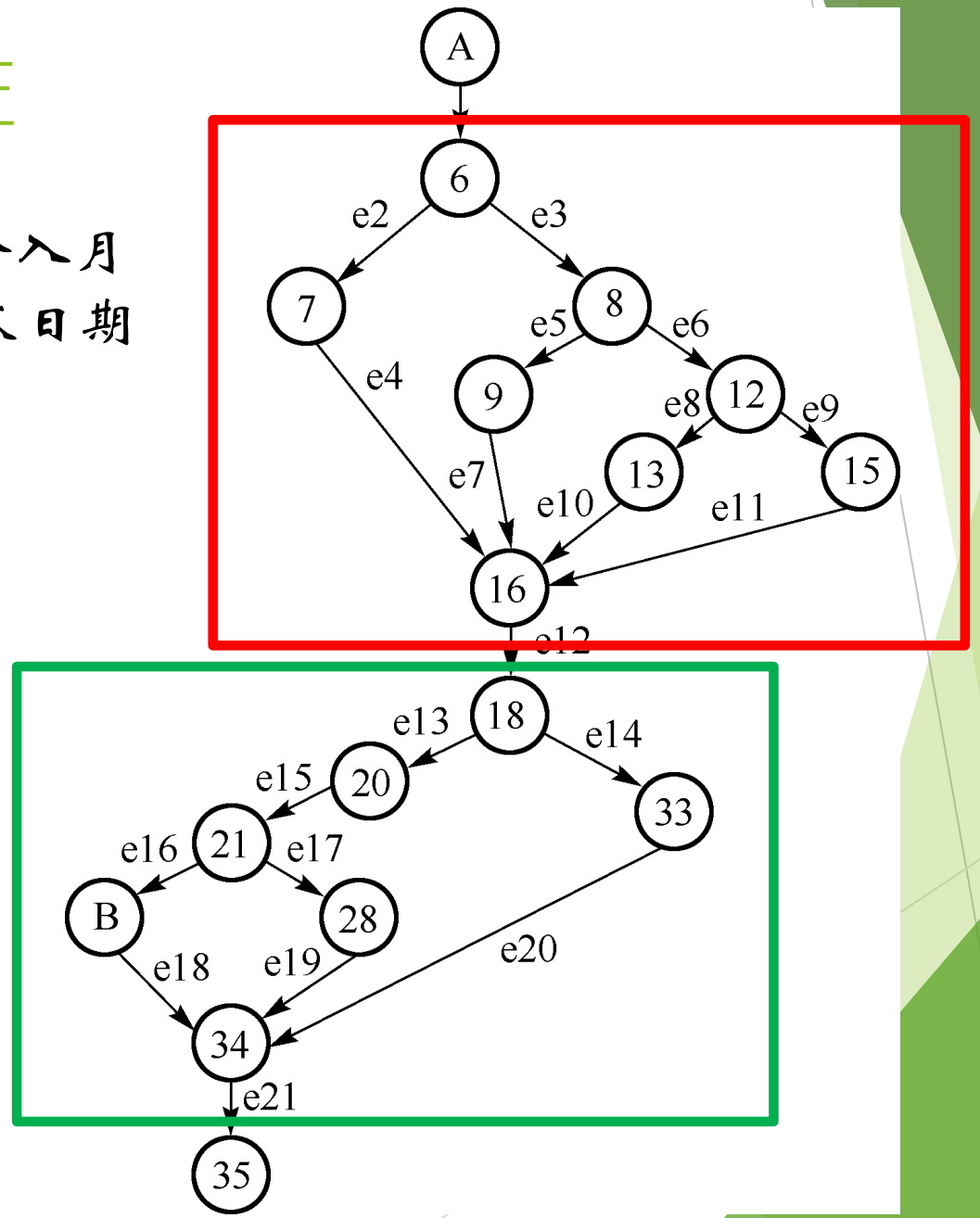
2. 如何处理不可行路径

- 结合源代码寻找独立路径
- 补充其他具有较高风险的路径进行测试

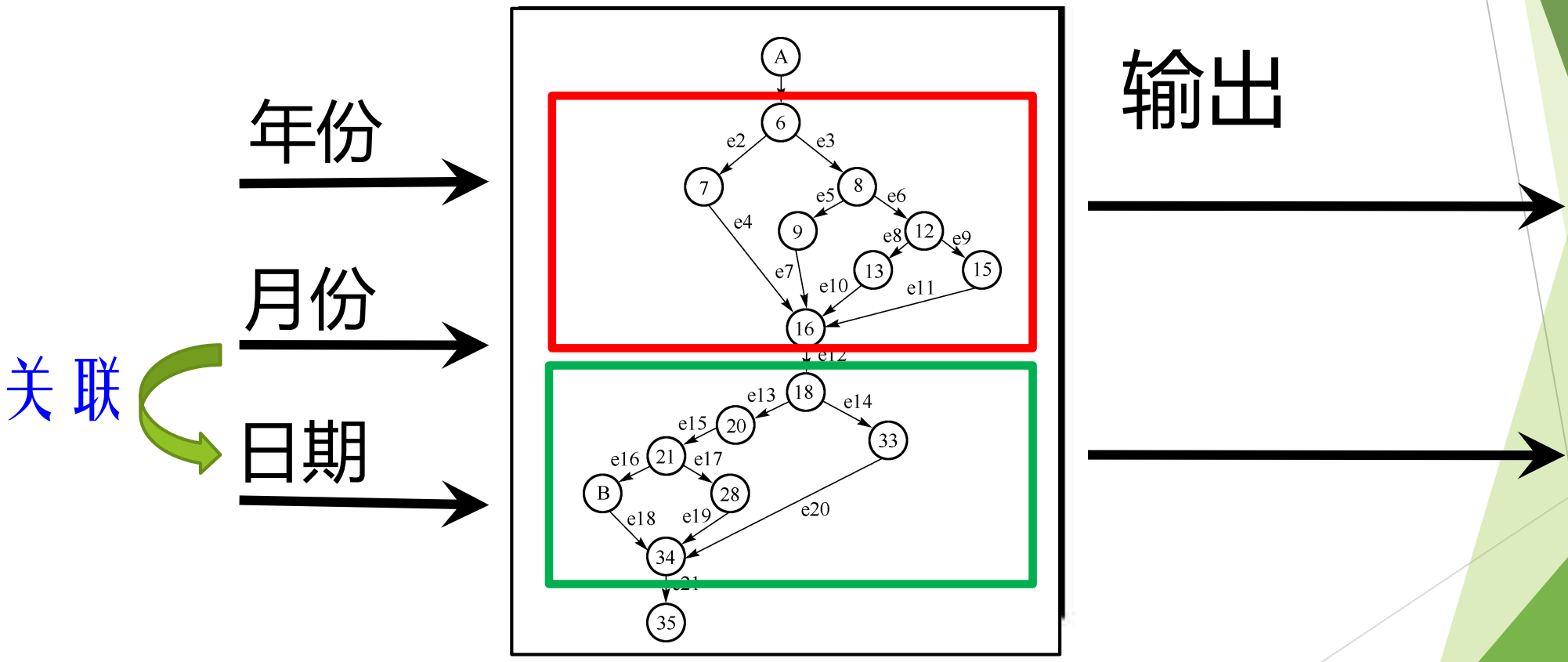
3. 不可行路径如何产生

2.确定输入月份的月末日期

3.判断日期类型并对应处理



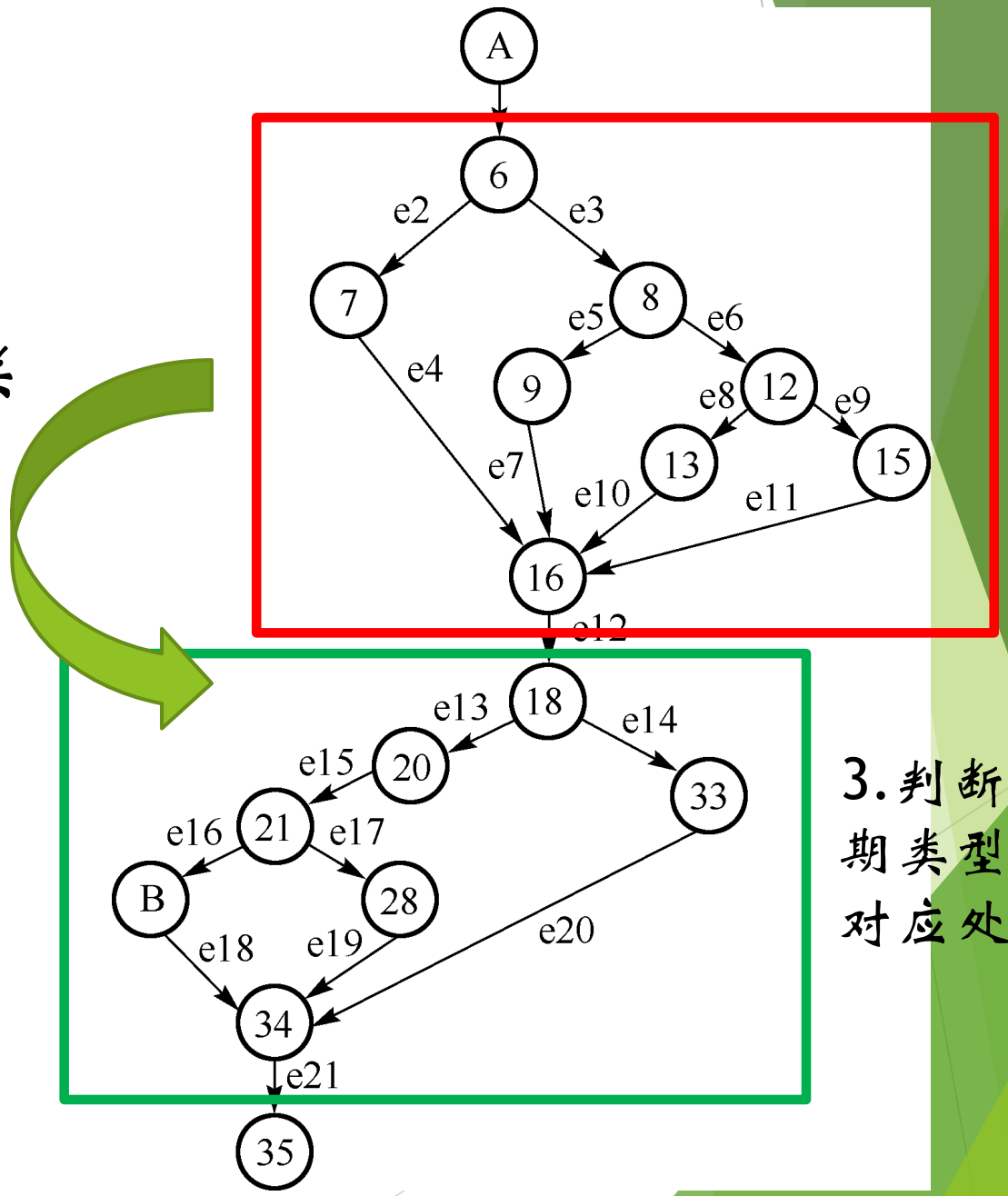
3. 不可行路径如何产生


年份
月份
日期
关联
输出
A
6
e2
e3
7
8
e5
e6
e4
9
12
e8
e9
e7
13
15
e10
e11
16
e12
18
e13
e14
e15
20
33
e16
21
e17
B
28
e18
e19
e20
34
e21
35

3. 不可行路径如何产生

- 判定结构体之间存在关联

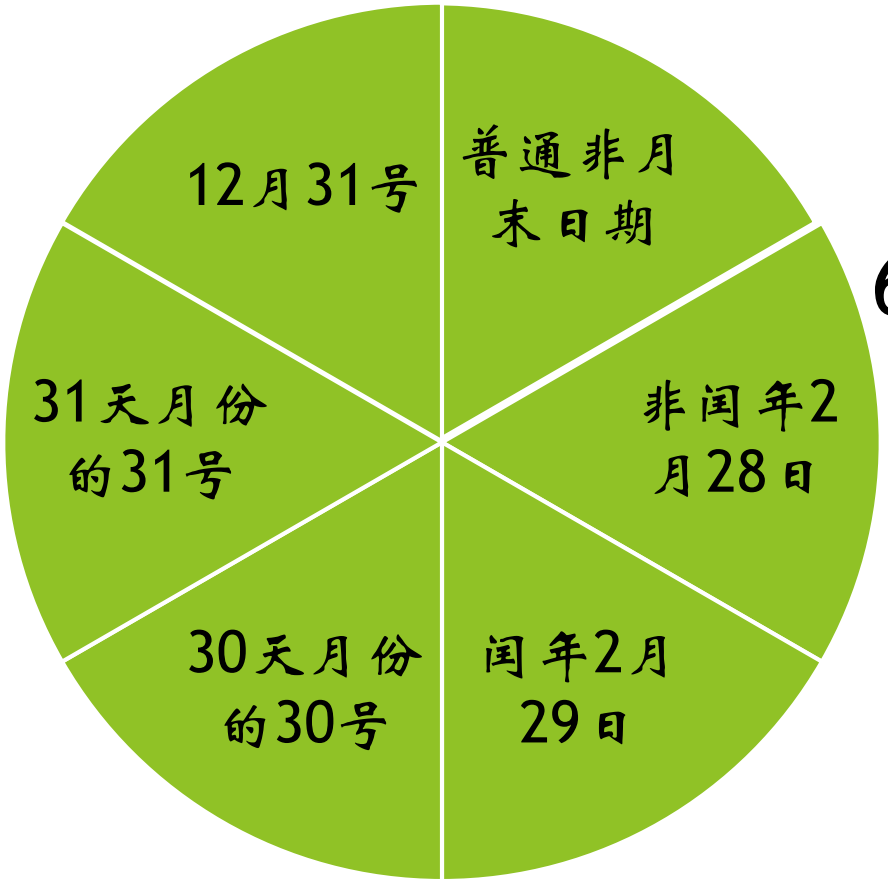
关联



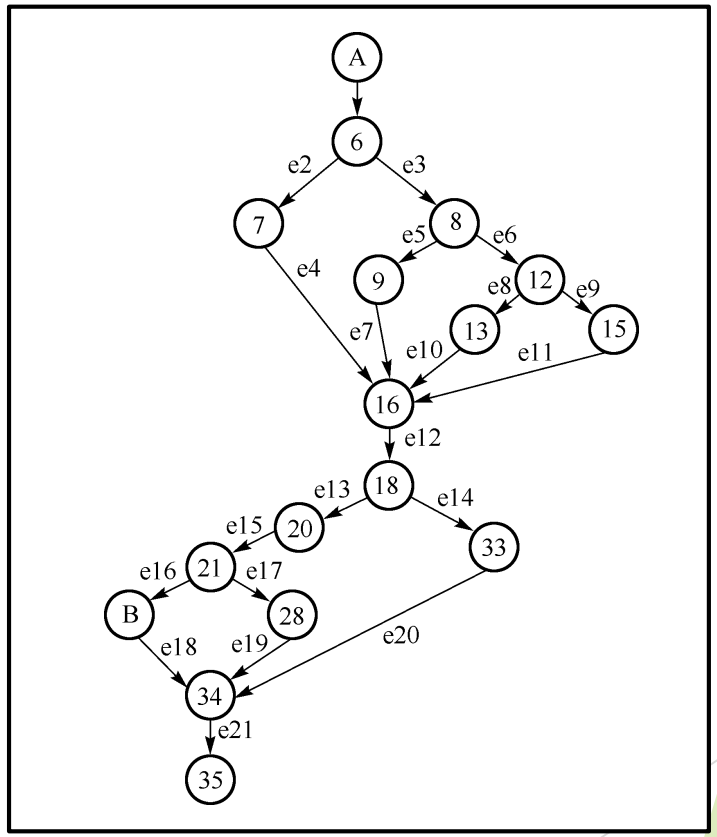
2.确定输入月份的月末日期

3.判断日期类型并对应处理

4. 开发过程中如何避免不可行路径



6条路径



输出

需求

程序实现

4. 开发过程中如何避免不可行路径

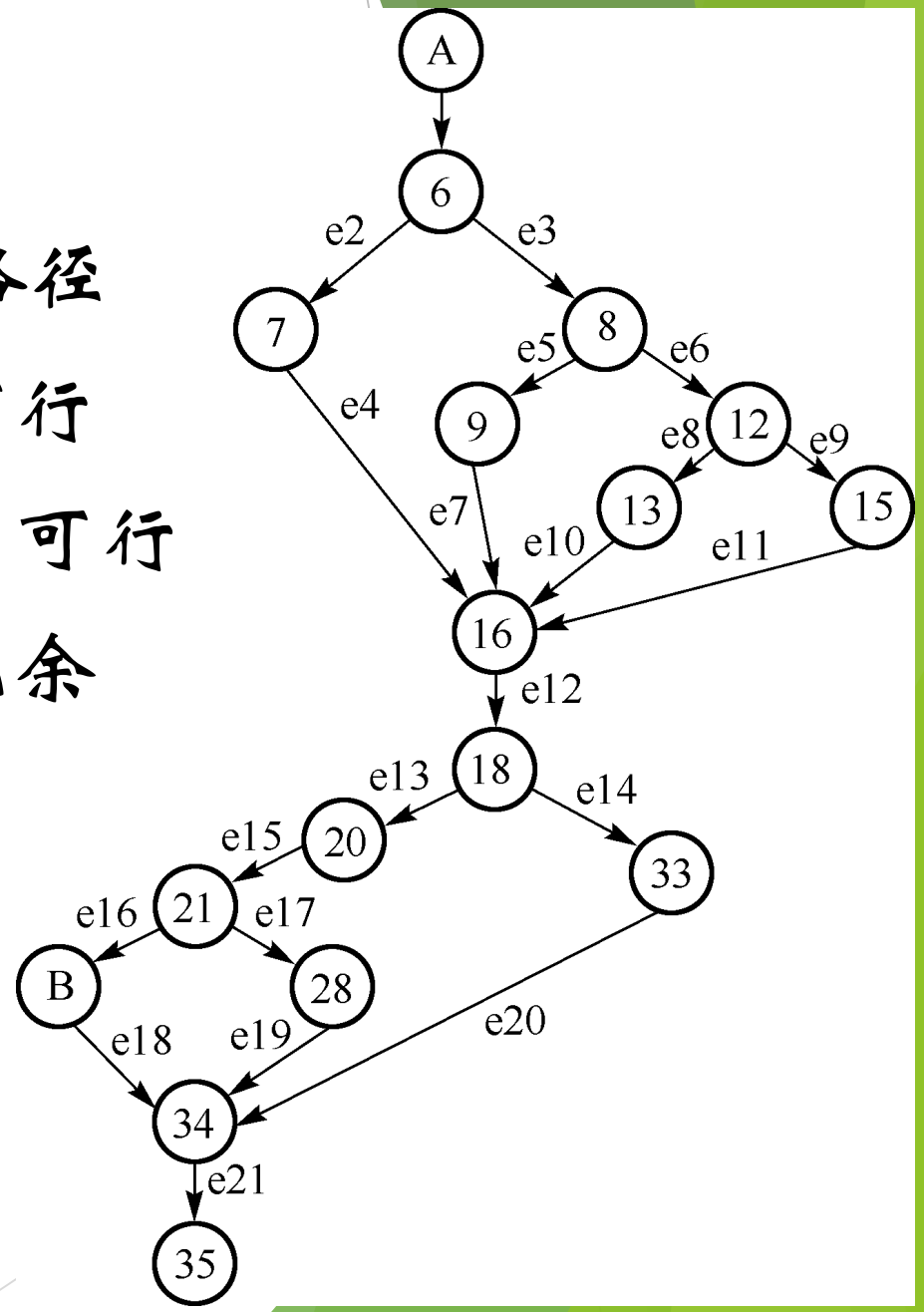
6条路径

- 普通非月末日期
- 非闰年2月28日
- 闰年2月29日
- 30天月份的30号
- 31天月份的31号
- 12月31号

需求与实现不一致

12条路径

- 6条可行
- 3条不可行
- 3条冗余



4. 开发过程中如何避免不可行路径

- 从测试的角度看待和分析需求
- 设计合理的程序结构