

# 1. 软件说明

**程序名称：**《南航国内/国际航班行李托运计算服务》

**程序介绍：**本程序接收行李信息，包括行李的重量、尺寸、旅客的舱位、VIP 等级、出发地和目的地，来计算行李的托运费。其中，行李重量和尺寸上限均取 45KG 和 300CM，国内航班超重按 15/KG 计算。

**程序说明：**程序由两部分组成，由 Golang 编写的使用 POST 请求的托运费计算服务和由 Python 编写的服务调用网站。

Golang 涉及的数据结构为四个结构体：

```
type Luggage struct {  
    Area int // 航线  
    Space int // 舱位  
    Vip int // VIP等级  
    Begin string // 出发地  
    End string // 到达地  
    Weights []int // 重量 切片  
    Sizes []int // 尺寸和 切片  
}
```

图 1 行李结构体

```
type OriginLuggage struct {  
    Space int // 舱位  
    Vip int // VIP等级  
    Begin string // 出发地  
    End string // 到达地  
    Weights []int // 重量 切片  
    Sizes [][]int // 二维尺寸 切片  
}
```

图 2 原始输入数据

```
type Limit struct {  
    number int // 免费行李件数上限  
    weight int // 单件免费行李重量上限  
    size int // 单件免费行李尺寸上限  
}
```

图 3 免费额度限制

```
type OverWeight struct {
    number int // 超额行李件数
    weights []int // 实际行李重量与免费额度的差值
    sizes []int // 实际行李尺寸与免费额度的差值
}
```

图 4 差值额度

对于 OverWeight 结构体，它的各项属性，值为正，说明超额的额度，值为负，说明当先项没有超额。

```
func calcFee(o OriginLuggage) int {
    var fee = -1
    if ok, _ := isLuggageValid(o); ok {
        luggage := origin2Luggage(o)
        fee = calc(luggage)
    }
    return fee
}
```

图 5 处理函数

处理函数仅接收一个 OriginLuggage 变量作为输入，isLuggagedValid 函数用来判断输入合法性，返回布尔值和错误信息，origin2Luggage 函数将原始数据转换为行李结构体，calc 根据行李结构体计算托运费。对于 calcFee，若返回值为 -1，则说明运行出错。

```

func calc(luggage Luggage) int {
    var fee int
    var limit Limit
    luggage = generateLuggage(luggage)
    sType := specialType(luggage)
    if sType > 0{
        luggage, limit = newSpecialLimit(luggage, sType)
    }
    switch luggage.Area {
        case area0:
            limit = newLimitArea0(luggage, limit)
            fee = calcArea0(luggage, limit)
        case area1:
            limit = newLimitArea1(luggage, limit)
            fee = calcArea1(luggage, limit)
        case area2:
            limit = newLimitArea2(luggage, limit)
            fee = calcArea2(luggage, limit)
        case area3:
            limit = newLimitArea3(luggage, limit)
            fee = calcArea3(luggage, limit)
        case area4:
            limit = newLimitArea4(luggage, limit)
            fee = calcArea4(luggage, limit)
    }
    return fee
}

```

图 6 运算函数

运算函数接收一个 Luggage 变量，generateLuggage 函数确定旅客的航线，specialType 函数检测特殊情况，newSpecialLimit 函数检测到特殊情况后对行李数据和免额额度数据进行修正，获取了行李数据和免额额度数据后根据航线将其放入对应的分支计算即可。

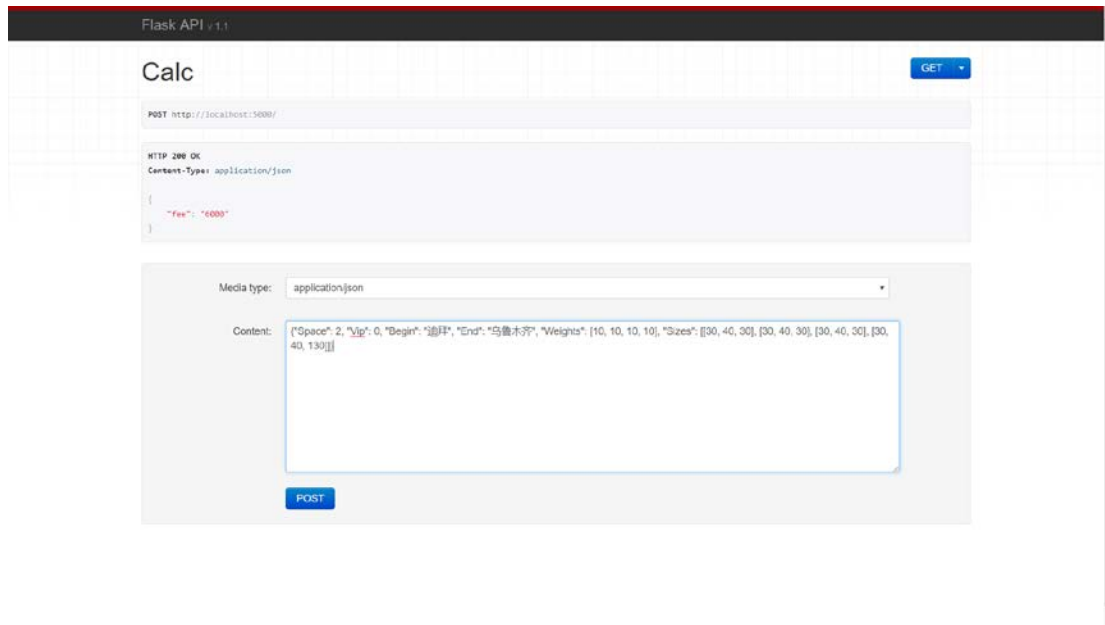


图 7 前端界面

服务调用如图所示，启动 main.go，启动 app.py，访问 localhost:5000，在 Content 输入框中输入 OriginLuggage 格式的 JSON 文本，再点击 POST 即可。（示例文本：{"Space": 2, "Vip": 0, "Begin": "迪拜", "End": "乌鲁木齐", "Weights": [10, 10, 10, 10], "Sizes": [[30, 40, 30], [30, 40, 30], [30, 40, 30], [30, 40, 130]]}）

## 2. 测试内容

测试用例根据如下程序流程图设计：

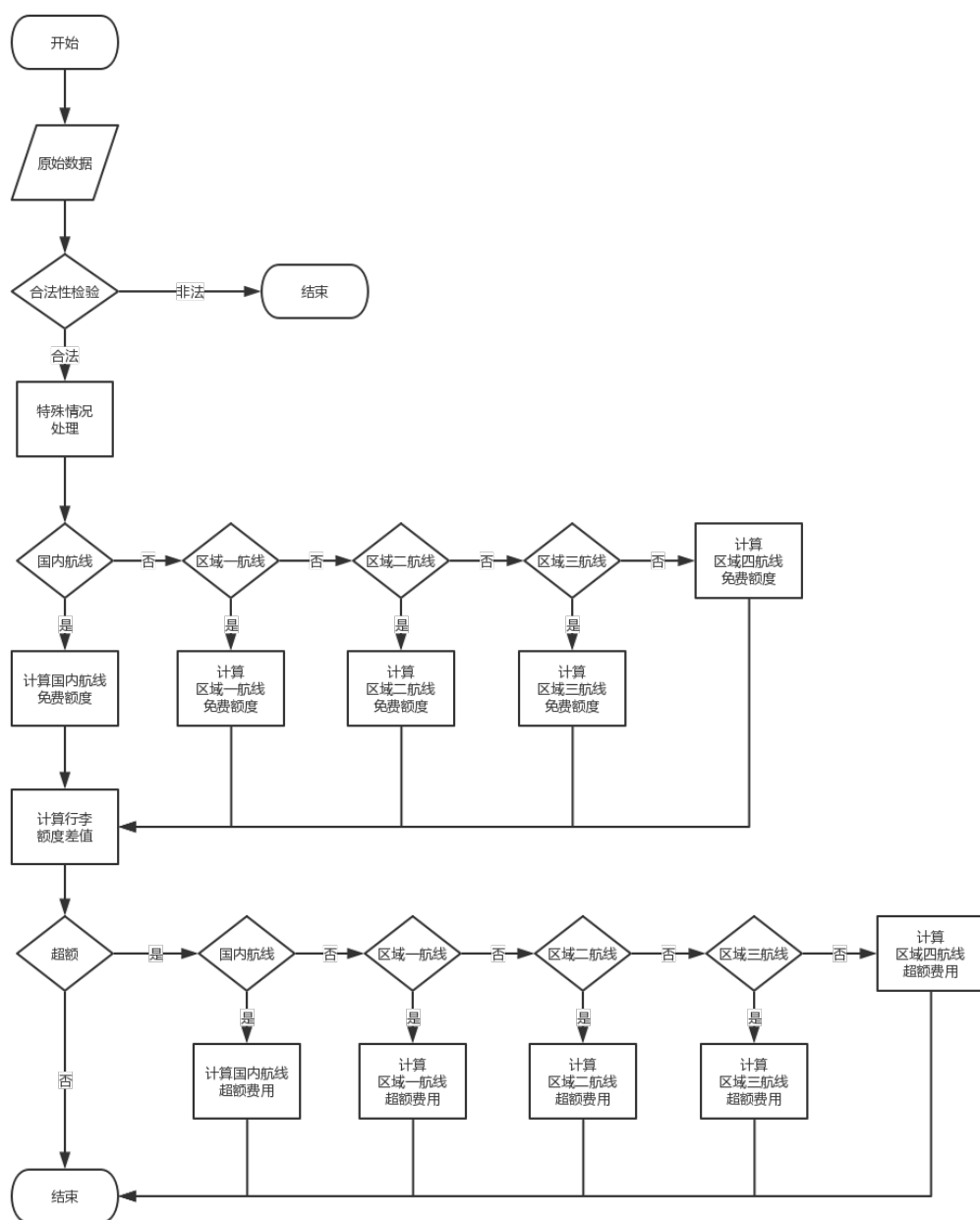


图 8 顶层流程图

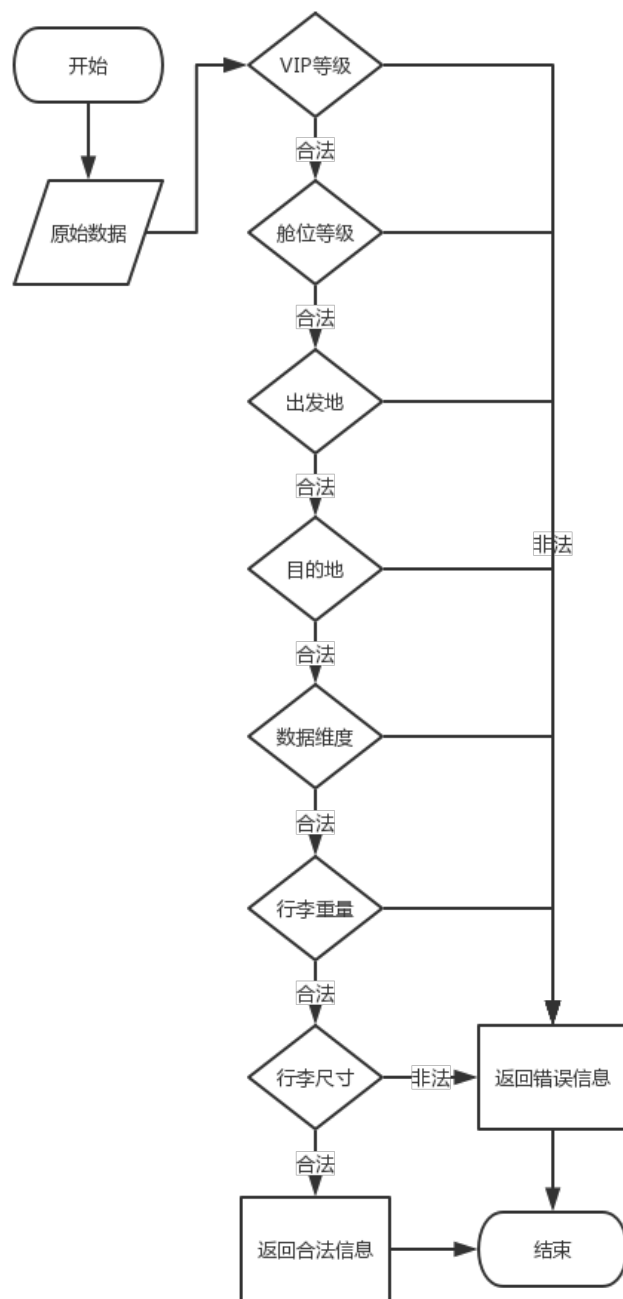


图 9 合法性检验流程图

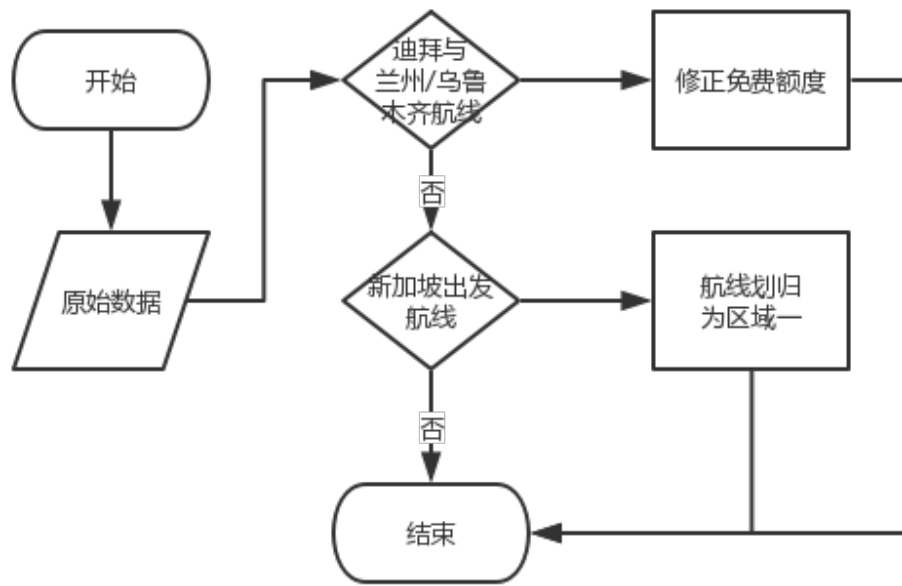


图 10 特殊情况处理流程图（韩国合并到了区域三）

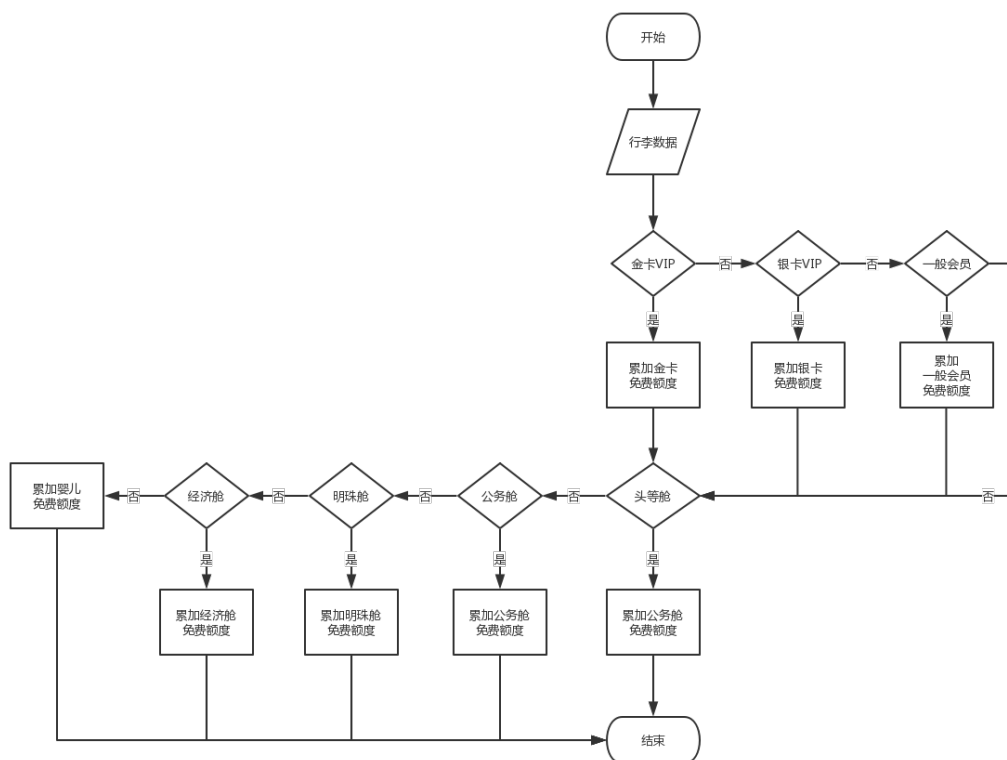


图 11 相关额度计算流程图

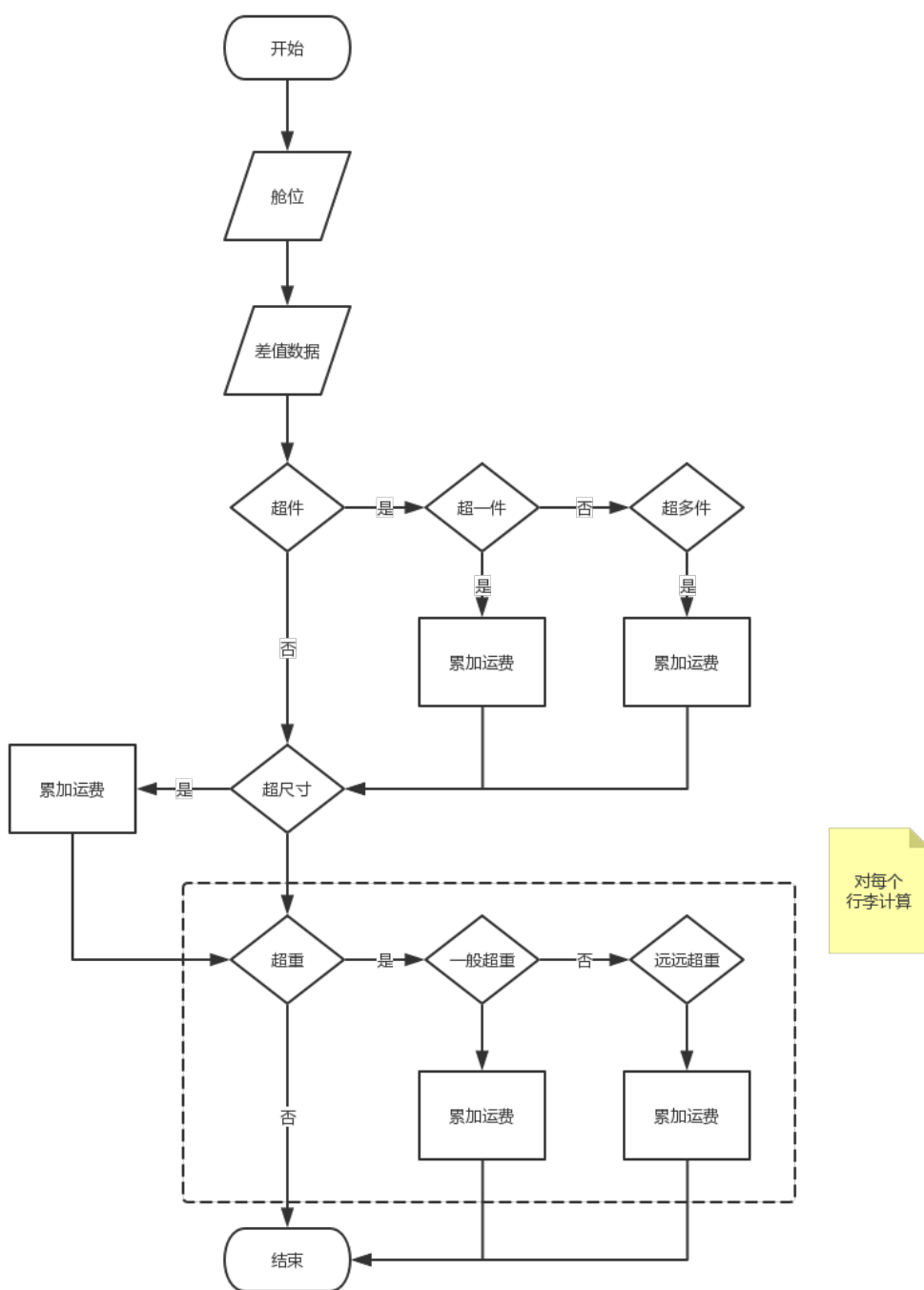


图 12 相关运费计算流程图



## 3. 测试用例设计说明

### 3.1 用例 1

#### 3.1.1 控制

会员等级小于会员等级范围

#### 3.1.2 输入

{Space:-5, Vip:-5, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[][int{20, 20, 20},  
[int{20, 20, 20}]}

#### 3.1.3 输出

"VIP Error"

#### 3.1.4 过程

输入检测->会员等级错误

### 3.2 用例 2

#### 3.2.1 控制

会员等级大于会员等级范围

#### 3.2.2 输入

{Space:-5, Vip:8, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[][int{20, 20, 20},  
[int{20, 20, 20}]}

#### 3.2.3 输出

"VIP Error"

### 3.2.4 过程

输入检测->会员等级错误

## 3.3 用例 3

### 3.3.1 控制

舱位等级大于舱位等级范围

### 3.3.2 输入

{Space:8, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[][int{int{20, 20, 20},  
[int{20, 20, 20}}]}

### 3.3.3 输出

"Space Error"

### 3.3.4 过程

输入检测->舱位等级检测

## 3.4 用例 4

### 3.4.1 控制

舱位等级小于舱位等级范围

### 3.4.2 输入

Space:-1, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[][int{int{20, 20, 20},  
[int{20, 20, 20}}]}

### 3.4.3 输出

"Space Error"

### 3.4.4 过程

输入检测->舱位等级检测

## 3.5 用例 5

### 3.5.1 控制

无出发地信息

### 3.5.2 输入

{Space:0, Vip:0, Begin:"", End:"", Weights:[int{10, 10}, Sizes:[[]int{[]int{20, 20, 20}, []int{20, 20, 20}}]}

### 3.5.3 输出

"No Begin"

### 3.5.4 过程

输入检测->无出发地信息

## 3.6 用例 6

### 3.6.1 控制

无目的地信息

### 3.6.2 输入

{Space:0, Vip:0, Begin:"武汉", End:"", Weights:[int{10, 10}, Sizes:[[]int{[]int{20, 20, 20}, []int{20, 20, 20}}]}

### 3.6.3 输出

"No End"

### 3.6.4 过程

输入检测->无目的地信息

## 3.7 用例 7

### 3.7.1 控制

数据维度不匹配

### 3.7.2 输入

```
{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{10}, Sizes:[][]int{[]int{20, 20, 20}, []int{20, 20, 20}}}
```

### 3.7.3 输出

"No Size Matching"

### 3.7.4 过程

输入检测->无维度匹配

## 3.8 用例 8

### 3.8.1 控制

尺寸数据缺失

### 3.8.2 输入

```
{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{10}, Sizes:[][]int{[]int{20, 20}}}
```

### 3.8.3 输出

"No Enough Data"

### 3.8.4 过程

输入检测->尺寸数据缺失

## 3.9 用例 9

### 3.9.1 控制

超过最大可携带限额

### 3.9.2 输入

{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{46, 10}, Sizes:[][int{int{46, 20, 20}, [int{20, 20, 20}}]}

### 3.9.3 输出

"Too Weight"

### 3.9.4 过程

输入检测->超过最大可携带限额

## 3.10 用例 10

### 3.10.1 控制

超过最大尺寸限额

### 3.10.2 输入

{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[][int{int{20, 20, 20}, [int{2000, 20, 20}}]}

### 3.10.3 输出

"Beyond Size"

#### 3.10.4 过程

输入检测->超过最大尺寸限额

### 3.11 用例 11

#### 3.11.1 控制

出现负数重量

#### 3.11.2 输入

{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{-10, 10}, Sizes:[][int{20, 20, 20},  
[int{20, 20, 20}]}

#### 3.11.3 输出

"Negative Weight"

#### 3.11.4 过程

输入检测->出现负数重量

### 3.12 用例 12

#### 3.12.1 控制

出现负数尺寸

#### 3.12.2 输入

{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[][int{20, 20, 20},  
[int{20, 20, -20}]}

#### 3.12.3 输出

"Negative Size"

#### **3.12.4 过程**

输入检测->出现负数尺寸

### **3.13 用例 13**

#### **3.13.1 控制**

国内航线，金卡会员，头等舱，超重

#### **3.13.2 输入**

{Vip:goldVip, Space:first, Begin:"武汉", End:"北京", Weights:[]int{45, 45, 45, 45}, Sizes:[]int{100, 100, 100, 100}}

#### **3.13.3 输出**

900

#### **3.13.4 过程**

国内航线->超额->超重

### **3.14 用例 14**

#### **3.14.1 控制**

国内航线，银卡会员，公务舱，未超额

#### **3.14.2 输入**

{Vip:silverVip, Space:business, Begin:"武汉", End:"北京", Weights:[]int{40, 40}, Sizes:[]int{100, 100}}, 0

#### **3.14.3 输出**

0

#### **3.14.4 过程**

国内航线->未超额

### **3.15 用例 15**

#### **3.15.1 控制**

国内航线，一般会员，明珠舱，超重

#### **3.15.2 输入**

{Vip:normalVIP, Space:bright, Begin:"武汉", End:"北京", Weights:[]int{21}, Sizes:[]int{100}}

#### **3.15.3 输出**

15

#### **3.15.4 过程**

国内航线->超额->超重

### **3.16 用例 16**

#### **3.16.1 控制**

国内航线，一般会员，经济舱，超重

#### **3.16.2 输入**

{Vip:normalVIP, Space:economy, Begin:"武汉", End:"北京", Weights:[]int{21}, Sizes:[]int{100}}

#### **3.16.3 输出**

15



#### 3.16.4 过程

国内航线->超额->超重

### 3.17 用例 17

#### 3.17.1 控制

国内航线，没有会员，婴儿，超重

#### 3.17.2 输入

{Vip:noVIP, Space:baby, Begin:"武汉", End:"北京", Weights:[int{40, 40}, Sizes:[int{100, 100}]}

#### 3.17.3 输出

1050

#### 3.17.4 过程

国内航线->超额->超重

### 3.18 用例 18

#### 3.18.1 控制

区域一航线，没有会员，头等舱，未超出免费额度

#### 3.18.2 输入

{Vip:noVIP, Space:first, Begin:"武汉", End:"日本", Weights:[int{32, 32, 32}, Sizes:[int{100, 100, 100}]}

#### 3.18.3 输出

0

### 3.18.4 过程

区域一航线->未超额

## 3.19 用例 19

### 3.19.1 控制

区域一航线，没有会员，头等舱，（超一件）

### 3.19.2 输入

{Vip:noVIP, Space:first, Begin:"武汉", End:"日本", Weights:[]int{32, 32, 10, 10}, Sizes:[]int{100, 100, 100, 100}}

### 3.19.3 输出

1000

### 3.19.4 过程

区域一航线->超额->（超一件）

## 3.20 用例 20

### 3.20.1 控制

区域一航线，没有会员，公务舱，（超尺寸，严重超重）

### 3.20.2 输入

{Vip:noVIP, Space:business, Begin:"武汉", End:"日本", Weights:[]int{32, 33}, Sizes:[]int{200, 100}}

### 3.20.3 输出

4000

### 3.20.4 过程

区域一航线->超额->（超尺寸，严重超重）

## 3.21 用例 21

### 3.21.1 控制

区域一航线，一般会员，明珠舱，（超多件）

### 3.21.2 输入

{Vip:normalVIP, Space:bright, Begin:" 武汉 ", End:" 日本 ", Weights:[]int{23, 23, 23, 23, 23},  
Sizes:[]int{100, 100, 100, 100, 100}}

### 3.21.3 输出

3000

### 3.21.4 过程

区域一航线->超额->（超多件）

## 3.22 用例 22

### 3.22.1 控制

区域一航线，银卡会员，经济舱，（超尺寸，超重，严重超重）

### 3.22.2 输入

{Vip:silverVip, Space:economy, Begin:"日本", End:"武汉", Weights:[]int{24, 33, 23}, Sizes:[]int{100,  
100, 200}}

### 3.22.3 输出

5000

#### 3.22.4 过程

区域一航线->超额->（超尺寸，超重，严重超重）

### 3.23 用例 23

#### 3.23.1 控制

区域一航线，金卡会员，婴儿，（超多件，超尺寸，超重，严重超重）

#### 3.23.2 输入

```
{Vip:goldVip, Space:baby, Begin:"武汉", End:"日本", Weights:[]int{23, 23, 24, 33}, Sizes:[]int{100, 100, 200, 100}}
```

#### 3.23.3 输出

8000

#### 3.23.4 过程

区域一航线->超额->（超多件，超尺寸，超重，严重超重）

### 3.24 用例 24

#### 3.24.1 控制

区域二航线，没有会员，头等舱，未超出额度

#### 3.24.2 输入

```
{Vip:noVIP, Space:first, Begin:"武汉", End:"格鲁吉亚", Weights:[]int{32, 32, 32}, Sizes:[]int{100, 100, 100}}
```

#### 3.24.3 输出

0

#### **3.24.4 过程**

区域二航线->未超额

### **3.25 用例 25**

#### **3.25.1 控制**

区域二航线，没有会员，头等舱，（超一件）

#### **3.25.2 输入**

{Vip:noVIP, Space:first, Begin:"武汉", End:"格鲁吉亚", Weights:[int{32, 32, 32, 32}, Sizes:[int{100, 100, 100, 100}]}

#### **3.25.3 输出**

450

#### **3.25.4 过程**

区域二航线->超额->（超一件）

### **3.26 用例 26**

#### **3.26.1 控制**

区域二航线，没有会员，公务舱，（超多件）

#### **3.26.2 输入**

{Vip:noVIP, Space:business, Begin:"武汉", End:"格鲁吉亚", Weights:[int{32, 32, 32, 32}, Sizes:[int{100, 100, 100, 100}]}

#### **3.26.3 输出**

1750

### 3.26.4 过程

区域二航线->超额->（超多件）

## 3.27 用例 27

### 3.27.1 控制

区域二航线，一般会员，明珠舱（超尺寸，严重超重）

### 3.27.2 输入

{Vip:normalVIP, Space:bright, Begin:"武汉", End:"格鲁吉亚", Weights:[int{33, 33}, Sizes:[int{200, 159}]}

### 3.27.3 输出

8000

### 3.27.4 过程

区域二航线->超额->（超尺寸，严重超重）

## 3.28 用例 28

### 3.28.1 控制

区域二航线，银卡会员，经济舱，（超多件，超尺寸，严重超重）

### 3.28.2 输入

{Vip:silverVip, Space:economy, Begin:"格鲁吉亚", End:"武汉", Weights:[int{32, 33, 10, 10}, Sizes:[int{200, 100, 100, 100}]}

### 3.28.3 输出

5750

### 3.28.4 过程

区域二航线->超额->（超多件，超尺寸，严重超重）

## 3.29 用例 29

### 3.29.1 控制

区域二航线，金卡会员，婴儿，（超多件，超尺寸，严重超重）

### 3.29.2 输入

{Vip:goldVip, Space:baby, Begin:"武汉", End:"格鲁吉亚", Weights:[int{32, 33, 10, 10},  
Sizes:[int{200, 100, 100, 100}]}

### 3.29.3 输出

8750

### 3.29.4 过程

区域二航线->超额->（超多件，超尺寸，严重超重）

## 3.30 用例 30

### 3.30.1 控制

区域三航线，没有会员，头等舱，未超出额度

### 3.30.2 输入

{Vip:noVIP, Space:first, Begin:"武汉", End:"内罗毕", Weights:[int{32, 32, 32}, Sizes:[int{100, 100,  
100}]}

### 3.30.3 输出

0

### 3.30.4 过程

区域三航线->未超额

## 3.31 用例 31

### 3.31.1 控制

区域三航线，没有会员，头等舱，（超一件）

### 3.31.2 输入

{Vip:noVIP, Space:first, Begin:"武汉", End:"内罗毕", Weights:[int{32, 32, 32, 32}, Sizes:[int{100, 100, 100, 100}]}

### 3.31.3 输出

1000

### 3.31.4 过程

区域三航线->超额->（超一件）

## 3.32 用例 32

### 3.32.1 控制

区域三航线，没有会员，公务舱，（超尺寸，严重超重）

### 3.32.2 输入

{Vip:noVIP, Space:business, Begin:"武汉", End:"内罗毕", Weights:[int{32, 33}, Sizes:[int{200, 100}]}

### 3.32.3 输出

4000



#### 3.32.4 过程

区域三航线->超额->（超尺寸，严重超重）

### 3.33 用例 33

#### 3.33.1 控制

区域三航线，一般会员，明珠舱，（超多件）

#### 3.33.2 输入

{Vip:normalVIP, Space:bright, Begin:"武汉", End:"内罗毕", Weights:[int{23, 23, 23, 23, 23},  
Sizes:[int{100, 100, 100, 100, 100}]}

#### 3.33.3 输出

3000

#### 3.33.4 过程

区域三航线->超额->（超多件）

### 3.34 用例 34

#### 3.34.1 控制

区域三航线，白银会员，经济舱，（超尺寸，超重，严重超重）

#### 3.34.2 输入

{Vip:silverVip, Space:economy, Begin:"内罗毕", End:"武汉", Weights:[int{32, 33}, Sizes:[int{200,  
100}]}

#### 3.34.3 输出

6000

#### 3.34.4 过程

区域三航线->超额->（超尺寸，超重，严重超重）

### 3.35 用例 35

#### 3.35.1 控制

区域三航线，金卡会员，婴儿，（超多件，超尺寸，超重，严重超重）

#### 3.35.2 输入

{Vip:goldVip, Space:baby, Begin:"武汉", End:"内罗毕", Weights:[int{32, 33, 10, 10}], Sizes:[int{200, 100, 100, 100}]}

#### 3.35.3 输出

9000

#### 3.35.4 过程

区域三航线->超额->（超多件，超尺寸，超重，严重超重）

### 3.36 用例 36

#### 3.36.1 控制

区域四航线，没有会员，头等舱，没有超出额度

#### 3.36.2 输入

{Vip:noVIP, Space:business, Begin:"武汉", End:"艾尔", Weights:[int{23, 23}], Sizes:[int{100, 100}]}

#### 3.36.3 输出

0

#### **3.36.4 过程**

区域四航线->未超额

### **3.37 用例 37**

#### **3.37.1 控制**

区域四航线，没有会员，头等舱，（超一件）

#### **3.37.2 输入**

{Vip:noVIP, Space:first, Begin:"武汉", End:"艾尔", Weights:[]int{32, 32, 32, 32}, Sizes:[]int{100, 100, 100, 100}}

#### **3.37.3 输出**

450

#### **3.37.4 过程**

区域四航线->超额->（超一件）

### **3.38 用例 38**

#### **3.38.1 控制**

区域四航线，没有会员，头等舱，（超尺寸，严重超重）

#### **3.38.2 输入**

{Vip:noVIP, Space:first, Begin:"武汉", End:"艾尔", Weights:[]int{32, 33, 23}, Sizes:[]int{200, 100, 100}}

#### **3.38.3 输出**

4000

### 3.38.4 过程

区域四航线->超额->（超尺寸，严重超重）

## 3.39 用例 39

### 3.39.1 控制

区域四航线，一般会员，明珠舱，（超多件）

### 3.39.2 输入

{Vip:normalVIP, Space:bright, Begin:" 武汉 ", End:" 艾尔 ", Weights:[]int{23, 23, 23, 23, 23},  
Sizes:[]int{100, 100, 100, 100, 100}}

### 3.39.3 输出

1750

### 3.39.4 过程

区域四航线->超额->（超多件）

## 3.40 用例 40

### 3.40.1 控制

区域四航线，银卡会员，经济舱，（超尺寸，超重，严重超重）

### 3.40.2 输入

{Vip:silverVip, Space:economy, Begin:" 艾尔 ", End:" 武汉 ", Weights:[]int{32, 33}, Sizes:[]int{200,  
100}}

### 3.40.3 输出

5000

#### 3.40.4 过程

区域四航线->超额->（超尺寸，超重，严重超重）

### 3.41 用例 41

#### 3.41.1 控制

区域四航线，银卡会员，公务舱，（超尺寸）

#### 3.41.2 输入

{Space:economy, Begin:"兰州", End:"迪拜", Weights:[]int{10, 10, 10, 10}, Sizes:[]int{100, 100, 100, 200}}

#### 3.41.3 输出

1000

#### 3.41.4 过程

区域四航线->超额->（超尺寸）

### 3.42 用例 42

#### 3.42.1 控制

区域四航线，金卡会员，婴儿，（超多件，超尺寸，超重，严重超重）

#### 3.42.2 输入

{Vip:goldVip, Space:baby, Begin:"武汉", End:"艾尔", Weights:[]int{32, 33, 10, 10, 23}, Sizes:[]int{200, 100, 100, 100, 100}}

#### 3.42.3 输出

8050

#### **3.42.4 过程**

区域四航线->超额->（超多件，超尺寸，超重，严重超重）

### **3.43 用例 43**

#### **3.43.1 控制**

特殊情况，没有会员，头等舱，（超一件，超尺寸）

#### **3.43.2 输入**

{Begin:"新加坡", End:"武汉", Weights:[int{10, 10, 10, 10}], Sizes:[int{158, 100, 100, 159}]}

#### **3.43.3 输出**

2000

#### **3.43.4 过程**

特殊情况->区域一航线->超额->（超一件，超尺寸）

### **3.44 用例 44**

#### **3.44.1 控制**

特殊情况，没有会员，经济舱，（超多件，超尺寸）

#### **3.44.2 输入**

{Space:economy, Begin:"兰州", End:"迪拜", Weights:[int{10, 10, 10, 10}], Sizes:[int{100, 100, 100, 200}]}

#### **3.44.3 输出**

6000

#### 3.44.4 过程

特殊情况->区域一航线->修正免费额度->超额->（超多件，超尺寸）

### 3.45 用例 45

#### 3.45.1 控制

特殊情况，没有会员，明珠舱，（超多件，超尺寸）

#### 3.45.2 输入

```
{Space:bright, Begin:"乌鲁木齐", End:"迪拜", Weights:[int{10, 10, 10, 10}, Sizes:[int{100, 100, 100, 200}]}
```

#### 3.45.3 输出

6000

#### 3.45.4 过程

特殊情况->区域一航线->修正免费额度->超额->（超多件，超尺寸）

### 3.46 用例 46

#### 3.46.1 控制

特殊情况，没有会员，头等舱，（超一件，超尺寸）

#### 3.46.2 输入

```
{Space:first, Begin:"迪拜", End:"乌鲁木齐", Weights:[int{10, 10, 10, 10}, Sizes:[int{100, 100, 100, 200}]}
```

#### 3.46.3 输出

2000

#### 3.46.4 过程

特殊情况->区域一航线->超额->（超一件，超尺寸）

### 3.47 用例 47

#### 3.47.1 控制

特殊情况，没有会员，公务舱，（超多件，超尺寸）

#### 3.47.2 输入

```
{Space:business, Begin:"迪拜", End:"兰州", Weights:[]int{10, 10, 10, 10}, Sizes:[]int{100, 100, 100, 200}}
```

#### 3.47.3 输出

4000

#### 3.47.4 过程

特殊情况->区域一航线->超额->（超多件，超尺寸）

### 3.48 用例 48

#### 3.48.1 控制

合法输入

#### 3.48.2 输入

```
{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{10, 10}, Sizes:[][]int{[]int{20, 20, 20}, []int{20, 20, 20}}}
```

#### 3.48.3 输出

""



#### 3.48.4 过程

输入检测->检测通过

### 3.49 用例 49

#### 3.49.1 控制

出现负数尺寸

#### 3.49.2 输入

```
{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[][int{20, 20, 20},  
[int{20, -20, 20}]}
```

#### 3.49.3 输出

" Negative Size "

#### 3.49.4 过程

输入检测->出现负数尺寸

### 3.50 用例 50

#### 3.50.1 控制

出现负数尺寸

#### 3.50.2 输入

```
{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[][int{20, 20, 20},  
[int{-20, 20, 20}]}
```

#### 3.50.3 输出

" Negative Size "

3.50.4 过程

输入检测->出现负数尺寸

4. 测试用例表

参照表 1-1、1-2

表 1-1 等价类划分表

序号	输入等价类	有效等价类	无效等价类
1	舱位	0(头等舱)/0 1(公务舱)/1 2(明珠舱)/2 3(经济舱)/3 4(婴儿)/4	<0/5 >4/6
2	VIP 等级	0(没有会员)/7 1(普通会员)/8 2(银卡会员)/9 3(金卡会员)/10	<0/11 >3/12
3	出发地	非空字符串/13	空字符串/14
4	到达地	非空字符串/15	空字符串/16
5	行李重量	(0, 45]/17	<=0/18 >45/19
6	行李尺寸	(0, 300]/20	a <= 0    b <= 0    c <= 0/21 a + b + c > 300/22
7	航线	0(国内航线)/23 1(区域一航线)/24 2(区域二航线)/25 3(区域三航线)/26 4(区域四航线)/27 新加坡出发/28 乌鲁木齐与迪拜/38 兰州与迪拜/39	无(航线为计算而来)
8	数据维度	行李重量和行李尺寸 在一维上的数量相等 且每组尺寸数量为 3/29	行李重量和行李尺寸 在一维上的数量不等 /30 每组尺寸数量不为 3/31
9	超额	超一件/32 超多件/33 超尺寸/34	行李处于 免费额度之内/37

		超重/35 严重超重/36	
--	--	------------------	--

说明：斜杠前面为等价类，斜杠后面为编号。对于 JSON 无法转换为相应结构体的情况，将由标准库抛出错误，处理函数将被跳过。

表 1-2 测试用例设计表

用例编号	用例操作描述	测试数据	期望结果	覆盖等价类
1	会员等级小于会员等级范围	{Space:-5, Vip:-5, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[int{int{20, 20, 20}, [int{20, 20, 20}}]}	"VIP Error"	5、11、13、15、17、20、23、29、37
2	会员等级大于会员等级范围	{Space:-5, Vip:8, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[int{int{20, 20, 20}, [int{20, 20, 20}}]}	"VIP Error"	5、12、13、15、17、20、23、29、37
3	舱位等级大于舱位等级范围	{Space:8, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[int{int{20, 20, 20}, [int{20, 20, 20}}]}	"Space Error"	6、7、13、15、17、20、23、29、37
4	舱位等级小于舱位等级范围	{Space:-1, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[int{int{20, 20, 20}, [int{20, 20, 20}}]}	"Space Error"	5、7、13、15、17、20、23、29、37
5	无出发地信息	Space:0, Vip:0, Begin:"", End:"", Weights:[int{10, 10}, Sizes:[int{int{20,	"No Begin"	0、7、14、15、17、20、23、29、37

		20, 20}, []int{20, 20, 20}}}		
6	无目的地信息	{Space:0, Vip:0, Begin:"武汉", End:"", Weights:[]int{10, 10}, Sizes:[]int{[]int{20, 20, 20}, []int{20, 20, 20}}}	"No End"	0、7、13、16、17、20、23、29、37
7	数据维度不匹配	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{10}, Sizes:[]int{[]int{20, 20, 20}, []int{20, 20, 20}}}	"No Size Matching"	0、7、13、15、17、20、23、30、37
8	尺寸数据缺失	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{10}, Sizes:[]int{[]int{20, 20}}}	"No Enough Data"	0、7、13、15、17、20、23、31、37
9	超过最大可携带限额	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{46, 10}, Sizes:[]int{[]int{46, 20, 20}, []int{20, 20, 20}}}	"Too Weight"	0、7、13、15、19、20、23、29、37
10	超过最大尺寸限额	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{10, 10}, Sizes:[]int{[]int{20, 20, 20}, []int{2000, 20, 20}}}	"Beyond Size"	0、7、13、15、17、22、23、29、37
11	出现非正重量	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{-10, 10},	"Negative Weight"	0、7、13、15、18、20、23、29、37

		Sizes:[[]int{[]int{20, 20, 20}, []int{20, 20, 20}}}		
12	出现非正尺寸	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[]int{10, 10}, Sizes:[[]int{[]int{20, 20, 20}, []int{20, 20, -20}}}	"Negative Size"	0、7、13、15、17、21、23、29、37
13	国内航线，金卡会员，头等舱，超重	{Vip:goldVip, Space:first, Begin:"武汉", End:"北京", Weights:[]int{45, 45, 45, 45}, Sizes:[]int{100, 100, 100, 100}}	900	(从本例到 47 例的所有用例均覆盖 13、15、17、20、29 等价类) 23、10、0、35
14	国内航线，银卡会员，公务舱，未超额	{Vip:silverVip, Space:business, Begin:"武汉", End:"北京", Weights:[]int{40, 40}, Sizes:[]int{100, 100}}, 0	0	23、9、1、37
15	国内航线，一般会员，明珠舱，超重	{Vip:normalVIP, Space:bright, Begin:"武汉", End:"北京", Weights:[]int{21}, Sizes:[]int{100}}	15	23、8、2、35
16	国内航线，一般会员，经济舱，超重	{Vip:normalVIP, Space:economy, Begin:"武汉", End:"北京", Weights:[]int{21}, Sizes:[]int{100}}	15	23、8、3、35
17	国内航线，没有会员，婴儿，超重	{Vip:noVIP, Space:baby, Begin:"武汉", End:"北京", Weights:[]int{40, 40}, Sizes:[]int{100, 40}}	1050	23、7、4、35

		100}}		
18	区域一航线， 没有会员，头等舱，未超出 免费额度	{Vip:noVIP, Space:first, Begin:" 武汉", End:"日本", Weights:[int{32, 32, 32}, Sizes:[int{100, 100, 100}}	0	24、7、0、37
19	区域一航线， 没有会员，头等舱，（超一 件）	{Vip:noVIP, Space:first, Begin:" 武汉", End:"日本", Weights:[int{32, 32, 10, 10}, Sizes:[int{100, 100, 100, 100}}	1000	24、7、0、32
20	区域一航线， 没有会员，公务舱，（超尺 寸，严重超 重）	{Vip:noVIP, Space:business, Begin:"武汉", End:" 日本", Weights:[int{32, 33}, Sizes:[int{200, 100}}	4000	24、7、1、 34、36
21	区域一航线， 一般会员，明珠舱，（超多 件）	{Vip:normalVIP, Space:bright, Begin:"武汉", End:" 日本", Weights:[int{23, 23, 23, 23}, Sizes:[int{100, 100, 100, 100, 100}}	3000	24、8、2、33
22	区域一航线， 银卡会员，经济舱，（超尺 寸，超重，严 重超重）	{Vip:silverVip, Space:economy, Begin:"日本", End:" 武汉", Weights:[int{24, 33, 23}, Sizes:[int{100, 100, 200}}	5000	24、9、3、 34、35、36
23	区域一航线， 金卡会员，婴儿，（超多 件，超尺寸， 超重，严重超 重）	{Vip:goldVip, Space:baby, Begin:" 武汉", End:"日本", Weights:[int{23, 23, 24, 33}, Sizes:[int{100, 100,	8000	24、10、4、 33、34、35、 36

		200, 100}}		
24	区域二航线， 没有会员，头等舱，未超出额度	{Vip:noVIP, Space:first, Begin:" 武汉", End:"格鲁吉 亚", Weights:[int{32, 32, 32}, Sizes:[int{100, 100, 100}}	0	25、7、0、37
25	区域二航线， 没有会员，头等舱，（超一件）	{Vip:noVIP, Space:first, Begin:" 武汉", End:"格鲁吉 亚", Weights:[int{32, 32, 32, 32}, Sizes:[int{100, 100, 100, 100}}	450	25、7、0、32
26	区域二航线， 没有会员，公务舱，（超多件）	{Vip:noVIP, Space:business, Begin:"武汉", End:" 格鲁吉亚", Weights:[int{32, 32, 32, 32}, Sizes:[int{100, 100, 100, 100}}	1750	25、7、1、33
27	区域二航线， 一般会员，明珠舱（超尺寸，严重超重）	{Vip:normalVIP, Space:bright, Begin:"武汉", End:" 格 鲁 吉 亚 ", Weights:[int{33, 33}, Sizes:[int{200, 159}}	8000	25、8、2、 34、36
28	区域二航线， 银卡会员，经济舱，（超多件，超尺寸，严重超重）	{Vip:silverVip, Space:economy, Begin:"格鲁吉亚", End:"武汉", Weights:[int{32, 33, 10, 10}, Sizes:[int{200, 100, 100, 100}}	5750	25、9、3、 33、34、36
29	区域二航线， 金卡会员，婴儿，（超多件，超尺寸，	{Vip:goldVip, Space:baby, Begin:" 武汉", End:"格鲁吉 亚",	8750	25、10、4、 33、34、36

	严重超重)	Weights:[int{32, 33, 10, 10}, Sizes:[int{200, 100, 100, 100}]}		
30	区域三航线， 没有会员，头等舱，未超出额度	{Vip:noVIP, Space:first, Begin:" 武汉", End:"内罗毕", Weights:[int{32, 32, 32}, Sizes:[int{100, 100, 100}]}	0	26、7、0、37
31	区域三航线， 没有会员，头等舱，（超一件）	{Vip:noVIP, Space:first, Begin:" 武汉", End:"内罗毕", Weights:[int{32, 32, 32, 32}, Sizes:[int{100, 100, 100, 100}]}	1000	26、7、0、32
32	区域三航线， 没有会员，公务舱，（超尺寸，严重超重）	{Vip:noVIP, Space:business, Begin:"武汉", End:" 内罗毕", Weights:[int{32, 33}, Sizes:[int{200, 100}]}	4000	26、7、1、34、36
33	区域三航线， 一般会员，明珠舱，（超多件）	{Vip:normalVIP, Space:bright, Begin:"武汉", End:" 内罗毕", Weights:[int{32, 33, 10, 10}, Sizes:[int{200, 100, 100, 100}]}	3000	26、8、2、33
34	区域三航线， 白银会员，经济舱，（超尺寸，超重，严重超重）	{Vip:silverVip, Space:economy, Begin:"内罗毕", End:"武汉", Weights:[int{32, 33}, Sizes:[int{200, 100}]}	6000	26、9、3、34、35、36
35	区域三航线， 金卡会员，婴儿，（超多件，超尺寸，	{Vip:goldVip, Space:baby, Begin:" 武汉", End:"内罗毕", Weights:[int{32,	9000	26、10、4、33、34、35、36



	超重, 严重超重)	33, 10, 10}, Sizes:[int{200, 100, 100, 100}		
36	区域四航线, 没有会员, 头等舱, 没有超出额度	{Vip:noVIP, Space:business, Begin:"武汉", End:" 艾尔", Weights:[int{23, 23}, Sizes:[int{100, 100}]}	0	27、7、0、37
37	区域四航线, 没有会员, 头等舱, (超一件)	{Vip:noVIP, Space:first, Begin:" 武汉", End:"艾尔", Weights:[int{32, 32, 32, 32}, Sizes:[int{100, 100, 100, 100}]}	450	27、7、0、32
38	区域四航线, 没有会员, 头等舱, (超尺寸, 严重超重)	{Vip:noVIP, Space:first, Begin:" 武汉", End:"艾尔", Weights:[int{32, 33, 23}, Sizes:[int{200, 100, 100}]}	4000	27、7、1、 34、36
39	区域四航线, 一般会员, 明珠舱, (超多件)	{Vip:normalVIP, Space:bright, Begin:"武汉", End:" 艾尔", Weights:[int{23, 23, 23, 23, 23}, Sizes:[int{100, 100, 100, 100, 100}]}	1750	27、8、2、33
40	区域四航线, 银卡会员, 经济舱, (超尺寸, 超重, 严重超重)	{Vip:silverVip, Space:economy, Begin:"艾尔", End:" 武汉", Weights:[int{32, 33}, Sizes:[int{200, 100}]}	5000	27、9、3、 34、35、36
41	区域四航线, 银卡会员, 公务舱, (超尺寸)	{Space:economy, Begin:"兰州", End:" 迪拜", Weights:[int{10, 10, 10, 10},	1000	27, 9, 1, 34

		Sizes:[int{100, 100, 100, 200}]		
42	区域四航线，金卡会员，婴儿，（超多件，超尺寸，超重，严重超重）	{Vip:goldVip, Space:baby, Begin:"武汉", End:"艾尔", Weights:[int{32, 33, 10, 10, 23}], Sizes:[int{200, 100, 100, 100, 100}]}	8050	27、10、4、33、34、35、36
43	特殊情况，没有会员，头等舱，（超一件，超尺寸）	{Begin:"新加坡", End:"武汉", Weights:[int{10, 10, 10, 10}], Sizes:[int{158, 100, 100, 159}]}	2000	28、7、0、32、34
44	特殊情况，没有会员，经济舱，（超多件，超尺寸）	{Space:economy, Begin:"兰州", End:"迪拜", Weights:[int{10, 10, 10, 10}], Sizes:[int{100, 100, 100, 200}]}	6000	39、7、3、33、34
45	特殊情况，没有会员，明珠舱，（超多件，超尺寸）	{Space:bright, Begin:"乌鲁木齐", End:"迪拜", Weights:[int{10, 10, 10, 10}], Sizes:[int{100, 100, 100, 200}]}	6000	38、7、2、33、34
46	特殊情况，没有会员，头等舱，（超一件，超尺寸）	{Space:first, Begin:"迪拜", End:"乌鲁木齐", Weights:[int{10, 10, 10, 10}], Sizes:[int{100, 100, 100, 200}]}	2000	38、7、0、32、34
47	特殊情况，没有会员，公务舱，（超多件，超尺寸）	{Space:business, Begin:"迪拜", End:"兰州", Weights:[int{10, 10, 10, 10}], Sizes:[int{100, 100, 100, 200}]}	4000	39、7、1、33、34

		Sizes:[int{100, 100, 100, 200}]		
48	合法输入	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[int{int{20, 20, 20}, [int{20, 20, 20}]}]}	""	0、7、13、15、17、20、23、29、37
49	出现非正尺寸	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[int{int{20, 20, 20}, [int{20, -20, 20}]}]}	"Negative Size"	0、7、13、15、17、21、23、29、37
50	出现非正尺寸	{Space:0, Vip:0, Begin:"武汉", End:"北京", Weights:[int{10, 10}, Sizes:[int{int{20, 20, 20}, [int{-20, 20, 20}]}]}	"Negative Size"	0、7、13、15、17、21、23、29、37

测试覆盖率 89.0%（除去非算法部分的覆盖率为 100%），覆盖可视化见 [coverage.html](#)。

因果图绘制：

原因：

- 1.会员等级
- 2.舱位等级
- 3.出发地
- 4.目的地
- 5.行李重量
- 6.行李尺寸

中间状态：

- 7.非法输入
- 8.航线
- 9.免费额度
- 10.实际行李量
- 11.差值额度
- 12.数据维度错误

结果：  
13.错误信息  
14.托运费用  
图示如下：

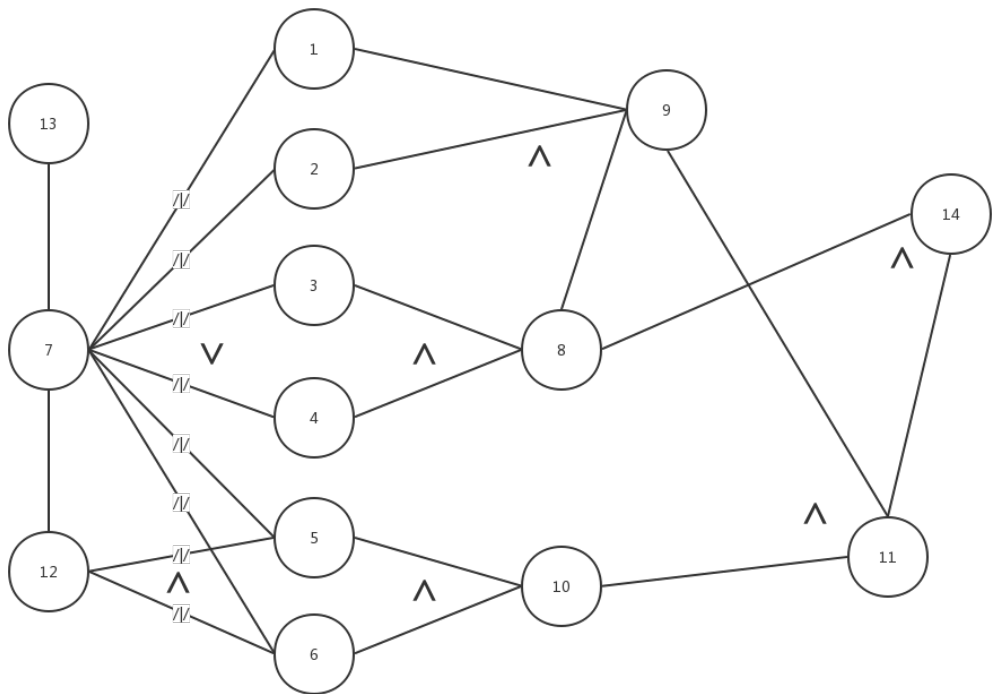


图 13 因果图

5. 测试结果

测试结果包括代码审查结果、测试用例的设计以及测试后的效果图。

表 2-2 测代码审查结果表

Bug ID	审查人员	审查日期	问题描述
BUG-001	吴梓杭	2019-3-18	区域一航线经济舱金卡会员严重超重行李划分到了超重行李
BUG-002	吴梓杭	2019-3-19	区域一航线经济舱银卡会员严重超重行李划分到了超重行李

附表一

测试用例执行结果统计表

测试项	测试用例号	测试特性	用例描述	测试结论	对应bug ID
输入数据格	1	数据合法性	会员等级小	passed	无

式			于会员等级范围		
输入数据格式	2	数据合法性	会员等级大于会员等级范围	passed	无
输入数据格式	3	数据合法性	舱位等级大于舱位等级范围	passed	无
输入数据格式	4	数据合法性	舱位等级小于舱位等级范围	passed	无
输入数据格式	5	数据合法性	无出发地信息	passed	无
输入数据格式	6	数据合法性	无目的地信息	passed	无
输入数据格式	7	数据合法性	数据维度不匹配	passed	无
输入数据格式	8	数据合法性	尺寸数据缺失	passed	无
输入数据格式	9	数据合法性	超过最大可携带限额	passed	无
输入数据格式	10	数据合法性	超过最大尺寸限额	passed	无
输入数据格式	11	数据合法性	出现非正重量	passed	无
输入数据格式	12	数据合法性	出现非正尺寸	passed	无
算法运算过程	13	逻辑正确性	国内航线，金卡会员，头等舱，超重	passed	无
算法运算过程	14	逻辑正确性	国内航线，银卡会员，公务舱，未超额	passed	无
算法运算过程	15	逻辑正确性	国内航线，一般会员，明珠舱，超重	passed	无
算法运算过程	16	逻辑正确性	国内航线，一般会员，经济舱，超重	passed	无

算法运算过程	17	逻辑正确性	国内航线，没有会员，婴儿，超重	passed	无
算法运算过程	18	逻辑正确性	区域一航线，没有会员，头等舱，未超出免费额度	passed	无
算法运算过程	19	逻辑正确性	区域一航线，没有会员，头等舱，（超一件）	passed	无
算法运算过程	20	逻辑正确性	区域一航线，没有会员，公务舱，（超尺寸，严重超重）	passed	无
算法运算过程	21	逻辑正确性	区域一航线，一般会员，明珠舱，（超多件）	passed	无
算法运算过程	22	逻辑正确性	区域一航线，银卡会员，经济舱，（超尺寸，超重，严重超重）	failed	BUG-001
算法运算过程	23	逻辑正确性	区域一航线，金卡会员，婴儿，（超多件，超尺寸，超重，严重超重）	failed	BUG-002
算法运算过程	24	逻辑正确性	区域二航线，没有会员，头等舱，未超出额度	passed	无
算法运算过程	25	逻辑正确性	区域二航	passed	无

程			线，没有会员，头等舱，（超一件）		
算法运算过程	26	逻辑正确性	区域二航线，没有会员，公务舱，（超多件）	passed	无
算法运算过程	27	逻辑正确性	区域二航线，一般会员，明珠舱（超尺寸，严重超重）	passed	无
算法运算过程	28	逻辑正确性	区域二航线，银卡会员，经济舱，（超多件，超尺寸，严重超重）	passed	无
算法运算过程	29	逻辑正确性	区域二航线，金卡会员，婴儿，（超多件，超尺寸，严重超重）	passed	无
算法运算过程	30	逻辑正确性	区域三航线，没有会员，头等舱，未超出额度	passed	无
算法运算过程	31	逻辑正确性	区域三航线，没有会员，头等舱，（超一件）	passed	无
算法运算过程	32	逻辑正确性	区域三航线，没有会员，公务舱，（超尺寸，严重超重）	passed	无
算法运算过程	33	逻辑正确性	区域三航	passed	无

程			线，一般会员，明珠舱，（超多件）		
算法运算过程	34	逻辑正确性	区域三航线，白银会员，经济舱，（超尺寸，超重，严重超重）	passed	无
算法运算过程	35	逻辑正确性	区域三航线，金卡会员，婴儿，（超多件，超尺寸，超重，严重超重）	passed	无
算法运算过程	36	逻辑正确性	区域四航线，没有会员，头等舱，没有超出额度	passed	无
算法运算过程	37	逻辑正确性	区域四航线，没有会员，头等舱，（超一件）	passed	无
算法运算过程	38	逻辑正确性	区域四航线，没有会员，头等舱，（超尺寸，严重超重）	passed	无
算法运算过程	39	逻辑正确性	区域四航线，一般会员，明珠舱，（超多件）	passed	无
算法运算过程	40	逻辑正确性	区域四航线，银卡会员，经济舱，（超尺寸，超重，严重超重）	passed	无



算法运算过程	41	逻辑正确性	区域四航线，银卡会员，公务舱，（超尺寸）	passed	无
算法运算过程	42	逻辑正确性	区域四航线，金卡会员，婴儿，（超多件，超尺寸，超重，严重超重）	passed	无
算法运算过程	43	逻辑正确性	特殊情况，没有会员，头等舱，（超一件，超尺寸）	passed	无
算法运算过程	44	逻辑正确性	特殊情况，没有会员，经济舱，（超多件，超尺寸）	passed	无
算法运算过程	45	逻辑正确性	特殊情况，没有会员，明珠舱，（超多件，超尺寸）	passed	无
算法运算过程	46	逻辑正确性	特殊情况，没有会员，头等舱，（超一件，超尺寸）	passed	无
算法运算过程	47	逻辑正确性	特殊情况，没有会员，公务舱，（超多件，超尺寸）	passed	无
数据输入格式	48	数据合法性	合法输入	passed	无
数据输入格式	49	数据合法性	出现非正尺寸	passed	无
数据输入格式	50	数据合法性	出现非正尺寸	passed	无