

# 离散优化建模：习题课 3

## 美人计：第一部分

### 1 问题描述

王允开始计划打败董卓。他设计让董卓跟他手下大将吕布同时爱上他的养女貂蝉。貂蝉有沉鱼落雁之容。当董卓被邀请到王允府上时，貂蝉准备了盛宴来引诱董卓。这场宴席就是美人计的第一部分。

貂蝉需要准备最丰盛的宴席。宴席上的菜必须精美和营养均衡，这样整个系列的菜式才称得上是丰盛。

貂蝉知道如何做一些菜。当然每道菜她只能上一次。

每一道菜的味道可以是 **辣**, **酸**, **咸**, **甜**, **鲜** 或者 **淡**。她不能连续上两道味道相同的菜。第一道菜必须是**咸**的，而最后一道菜必须是**甜**的。在每一道**辣**的菜之后必须有一道**淡**或者**甜**的菜。在每一道**酸**的菜之后必须有一道**淡**或者**鲜**味道的菜。在**甜**味菜之后不可以马上上**辣**或者**鲜**味的菜。

每一道菜可以是**热菜**, **凉菜** 或者**温菜**。在上了一道**热菜**之后，一道**凉菜**之前，必须有一道**温菜**。

每一道菜可以是难消化的或者易消化的。貂蝉准备的菜是种不可以连续有两道难消化的菜。

她的目标是令到菜式最好吃。而是否好吃取决于每道菜的得分加上连续的两道菜之间味道，温度和是否易消化等特征的变化次数。

### 2 数据格式说明

宴席问题的输入是名为 `data/feasttrap_*.dzn` 的数据文件，其中  $p$  是问题的序号。 $len$  是宴会菜式的数量。 $DISH$  是一些菜式的集合。 $taste$  是一个数组，代表不同的菜式的味道。 $temp$  是一个数组，代表不同菜式的温度。 $heavy$  是一个布尔型数组，表示每一道菜是否难以消化。 $value$  表示每一道菜的得分。决策变量是  $dish$ ，表示一个长为  $len$  的一维数组，代表每一道菜对应的菜式。

所以数据和决策变量的声明如下：

```
enum DISH;  
enum TASTE = {spicy, sour, salty, sweet, urname, bland};  
enum TEMP = {hot, cold, warm};  
  
array[DISH] of TASTE: taste;  
array[DISH] of TEMP: temp;  
array[DISH] of bool: heavy;
```

```
array[DISH] of int: value;
```

```
int: len; % length of banquet  
set of int: COURSE = 1..len;
```

```
array[COURSE] of var DISH: dish;
```

一个样本数据文件是:

```
len = 6;
```

```
DISH = { MAPOTOFU, KUNGPAOCHICKEN, COCONUTJELLY, LAIWONGBAO, CHARSIUBAO,  
          SESAMEPRAWN, HOTSOURSOUP, CHILIDUMPLINGS, GLASSNOODLES, FRIEDRICE };
```

```
taste = [umame, spicy, sweet, sweet, salty, salty, sour, spicy, bland, bland];  
temp = [ hot, hot, warm, cold, warm, warm, hot, hot, cold, warm];  
heavy = [ true, true, false, false, false, true, true, true, false, false];  
value = [ 8, 4, 5, 3, 5, 7, 4, 3, 2, 1 ];
```

这里面从 10 种菜式里面选取了宴会上的 6 道菜。你的模型的输出需要给出决策变量和目标数值。例如它可以输出:

```
dish = [CHARSIUBAO, LAIWONGBAO, GLASSNOODLES, SESAMEPRAWN, MAPOTOFU, COCONUTJELLY];  
obj = 41;
```

### 3 软件需求

你需要 MINIZINC 2.0 (<http://www.minizinc.org/2.0/>) 来完成这次习题课。





