FIT5216: Modelling Discrete Optimization Problems
Inclass Task 11: Voucher

1 Problem Statement

The Voucher problem is defined as follows. You have a number (m) of vouchers of the form: buy *buy* pizzas and get *free* pizzas (of no greater price) for free. Given a set of n pizzas $PIZZA$ you need to buy, each with a price $price[p], p \in PIZZA$, how should you use vouchers to get all the pizzas for the minimal price. You can't use a bought pizza to "enable" two different vouchers.

Luckily you are given a MiniZinc model: `voucher.mzn`. The main decision variables in the model are called `how`. For each pizza p, `how[p]` determines which voucher, if any, is used to obtain that pizza. If `how[p]` is a positive number i, it means you get that pizza for free using voucher i. If `how[p]` is a negative number $-i$, you pay for pizza p and use it to "unlock" voucher i. If it is 0, it simply means that you pay for the pizza and don't use it with any voucher.

Unfortunately the model has bugs. For the simple data file of 5 pizzas with prices 12, 25, 17, 12, and 9 and two vouchers, buy 1 get 1 free, and buy 2 get 1 free, defined by data file:

```
n = 5;
price = [12,25,17,12,9];
m = 2;
buy = [1,2];
free = [1,1];
```

the model returns the answer

```
how = [1, 1, 1, 1, 1];
cost = 0;
```

which seems to indicate that we can get all the pizzas for free.

Track down the bugs in the model, and fix them.

If you are up to the challenge, see if you can build a totally different model that is capable of rapidly finding the optimal solution 564039 for data `v2.dzn`.