## Constraint Programming

I believe that basically all the "CLP" stuff you need to use for at least the first eight problems are ::, the # constraints(#=,##,#>,#<,#>=,#=<), minimize, labeling, and alldifferent.

1. "Mrs. Spooner called this morning," said the honest grocer to his assistant. "She wants twenty pounds of tea at 2s. 4½d. per lb. Of course we have a good 2s. 6d. tea, a slightly inferior at 2s. 3d., and a cheap Indian at 1s. 9d., but she is very particular always about her prices."

"What do you propose to do?" asked the innocent assistant.

"Do?" exclaimed the grocer. "Why, just mix up the three teas in different proportions so that the twenty pounds will work out fairly at the lady's price. Only don't put in more of the best tea than you can help, as we make less profit on that, and of course you will use only our complete pound packets. Don't do any weighing."

How was the poor fellow to mix the three teas? Could eclipse have shown him how to do it? (Hint: 1s = 12d one shilling is 12 pence. You want to rescale to avoid fractions.)

2. A man bought an odd lot of wine in barrels and one barrel containing beer. These are shown in the illustration, marked with the number of gallons that each barrel contained. He sold a quantity of the wine to one man and twice the quantity to another, but kept the beer to himself. The puzzle is to point out which barrel contains beer. Can you say which one it is? Of course, the man sold the barrels just as he bought them, without manipulating in any way the contents.


3. The nine digits 1-9 are used to form one three-digit and three two-digit numbers so that the product of the three digit number and the first two-digit number is the same as the product of the second and third two-digit numbers. E.g. 158 multiplied by 23 is 3,634, and 79 multiplied by 46 is also 3,634. Write an eclipse program to find all such arrangements of digits, and one that will find the arrangement with the largest product. (Hint: maximizing X is the same as minimizing -X)


4. A woman was carrying a basket of eggs to market when a passer-by bumped her. She dropped the basket and all the eggs broke. The passer-by, wishing to pay for her loss, asked, 'How many eggs were in your basket?'

'I don't remember exactly,' the woman replied, 'but I do recall that whether I divided the eggs by 2,3,4,5 or 6 there was always one egg left over. When I took the eggs out in groups of seven, I emptied the basket.' Write an eclipse constraint program to find the least possible number of eggs.

LCM

NO 111, 119 10 pairs

$\frac{N!}{r!(n-r)!}$ $\frac{5!}{2!\,3!} = 10$

5. Farmer Tompkins had five trusses of hay, which he told his man Hodge to weigh before delivering them to a customer. The stupid fellow weighed them two at a time in all possible ways, and informed his master that the weights in pounds were 110, 112, 113, 114, 115, 116, 117, 118, 120, and 121. Now, how was Farmer Tompkins to find out from these figures how much every one of the five trusses weighed singly? The reader may at first think that he ought to be told "which pair is which pair," or something of that sort, but it is quite unnecessary. Can you give the five correct weights?

1150

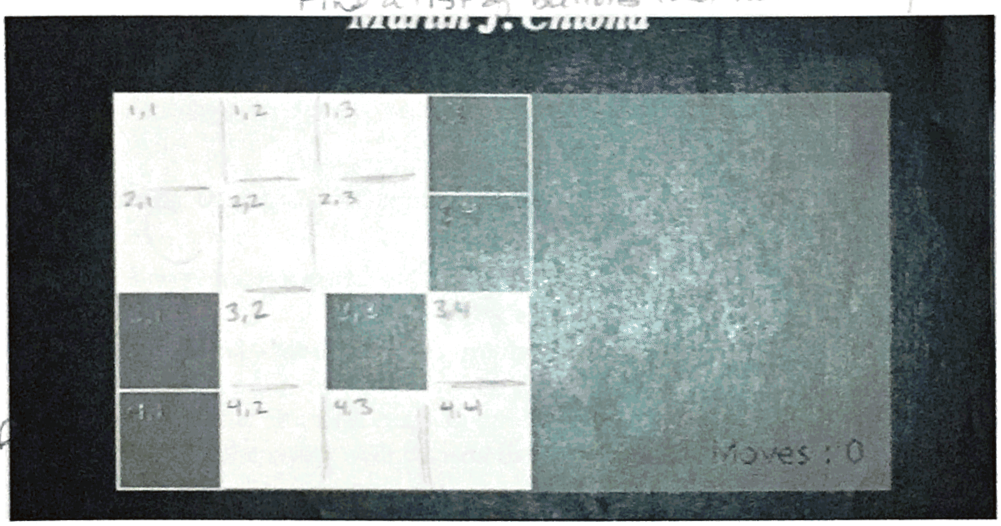
$\frac{5 \cdot 4}{2!} = 10$

6. Each of the squares in the grid below can be in one of two states, lit(white) or unlit(red). If the player clicks on a square then that square and all squares in the same row and column will toggle between the two states. Each mouse click constitutes one move and the objective of the puzzle is to light all 16 squares. The order doesn't matter

Find a list of buttons that need to be pressed

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Even # or odd # of buttons in that row/column have been pressed




(Hint: it is never necessary to click a square twice.)

## not equal

7. (from New Scientist magazine, 17 June 2006.) by W. Haigh.

I have allocated distinct positive integers to the letters of the alphabet. By adding up the values of the letters in their names, I have obtained the following scores for some members of the solar system: PLUTO 40, URANUS 36, NEPTUNE 29, SATURN 33, JUPITER 50, MARS 32, EARTH 31, MOON 36, VENUS 39, MERCURY 33, SUN 18.
Find the value of PLANETS. (You will find multiple solutions. Modify the problem and your solution to leave just one.) [PLUTORANSEJIMHVCY]

8. A farmer leaves 45 casks of wine, of which 9 each are full, three-quarters full, half full, one quarter full and empty. His five nephews want to divide the wine and the casks without changing wine from cask to cask in such a way that each receives the same amount of wine and the same number of casks, and further so that each receives at least one of each kind of cask, and no two of them receive the same number of every kind of cask.

9. Redo the party problem as constraint program.

Submit these to CS231Constraint3

Note: In place of labeling, it is also possible to get more control by using search/6. If you want to label the variables in the list VARS, you can do this:

search(Vars, 0, input_order, indomain, complete, [])

The third parameter can be varied, with the following options:

The pre-defined **selection methods** are the following:

- **input_order** the first entry in the list is selected
- **first_fail** the entry with the smallest domain size is selected
- **anti_first_fail** the entry with the largest domain size is selected
- **smallest** the entry with the smallest value in the domain is selected
- **largest** the entry with the largest value in the domain is selected
- **occurrence** the entry with the largest number of attached constraints is selected
- **most_constrained** the entry with the smallest domain size is selected. If several entries have the same domain size, the entry with the largest number of attached constraints is selected.
- **max_regret** the entry with the largest difference between the smallest and second smallest value in the domain is selected. This method is typically used if the variable represents a cost, and we are interested in the choice which could increase overall cost the most if the best possibility is not taken. Unfortunately, the implementation does not always work: If two decision variables incur the same minimal cost, the regret is not calculated as zero, but as the difference from this minimal value to the next greater value.

(There are lots of other options for other arguments as well.)