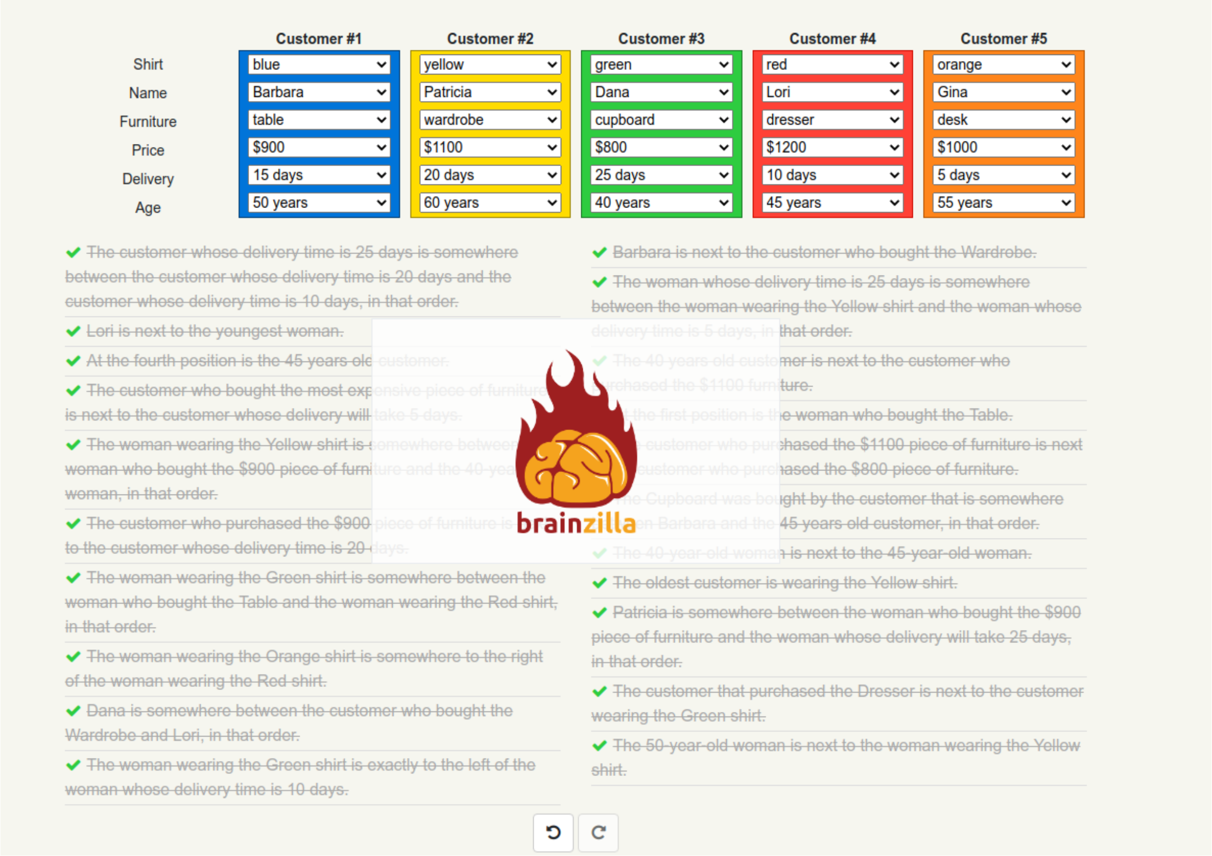
Assignments week 3

Aleksandar Ivanov, Ivaylo Ivanov

16.12.2020

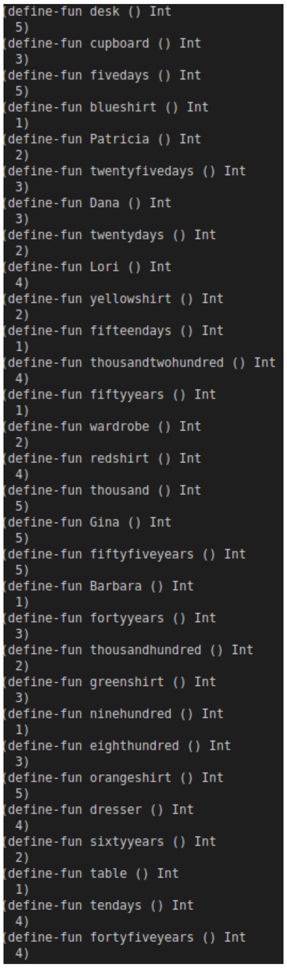
Assignment 1:

This assignment was very straightforward. The solution can be found in wk3_1.smt2. The question regarding each assertion is placed above it. We made 2 functions for the assertions that are often repeated, namely the “B is between A and C” and ”D is next to E” type of assertions. Below you can see the z3 output and the puzzle correctness proof.



	Customer #1	Customer #2	Customer #3	Customer #4	Customer #5
Shirt	blue	yellow	green	red	orange
Name	Barbara	Patricia	Dana	Lori	Gina
Furniture	table	wardrobe	cupboard	dresser	desk
Price	$900	$1100	$800	$1200	$1000
Delivery	15 days	20 days	25 days	10 days	5 days
Age	50 years	60 years	40 years	45 years	55 years

- ✓ ~~The customer whose delivery time is 25 days is somewhere between the customer whose delivery time is 20 days and the customer whose delivery time is 10 days, in that order.~~
- ✓ ~~Lori is next to the youngest woman.~~
- ✓ ~~At the fourth position is the 45 years old customer.~~
- ✓ ~~The customer who bought the most expensive piece of furniture is next to the customer whose delivery will take 5 days.~~
- ✓ ~~The woman wearing the Yellow shirt is somewhere between woman who bought the $900 piece of furniture and the 40-yea woman, in that order.~~
- ✓ ~~The customer who purchased the $900 piece of furniture is to the customer whose delivery time is 20 days.~~
- ✓ ~~The woman wearing the Green shirt is somewhere between the woman who bought the Table and the woman wearing the Red shirt, in that order.~~
- ✓ ~~The woman wearing the Orange shirt is somewhere to the right of the woman wearing the Red shirt.~~
- ✓ ~~Dana is somewhere between the customer who bought the Wardrobe and Lori, in that order.~~
- ✓ ~~The woman wearing the Green shirt is exactly to the left of the woman whose delivery time is 10 days.~~
- ✓ ~~Barbara is next to the customer who bought the Wardrobe.~~
- ✓ ~~The woman whose delivery time is 25 days is somewhere between the woman wearing the Yellow shirt and the woman whose delivery time is 5 days, in that order.~~
- ✓ ~~The 40 years old customer is next to the customer who purchased the $1100 furniture.~~
- ~~the first position is the woman who bought the Table.~~
- ~~e customer who purchased the $1100 piece of furniture is next customer who purchased the $800 piece of furniture.~~
- ~~he Cupboard was bought by the customer that is somewhere en Barbara and the 45 years old customer, in that order.~~
- ✓ ~~The 40-year-old woman is next to the 45-year-old woman.~~
- ✓ ~~The oldest customer is wearing the Yellow shirt.~~
- ✓ ~~Patricia is somewhere between the woman who bought the $900 piece of furniture and the woman whose delivery will take 25 days, in that order.~~
- ✓ ~~The customer that purchased the Dresser is next to the customer wearing the Green shirt.~~
- ✓ ~~The 50-year-old woman is next to the woman wearing the Yellow shirt.~~

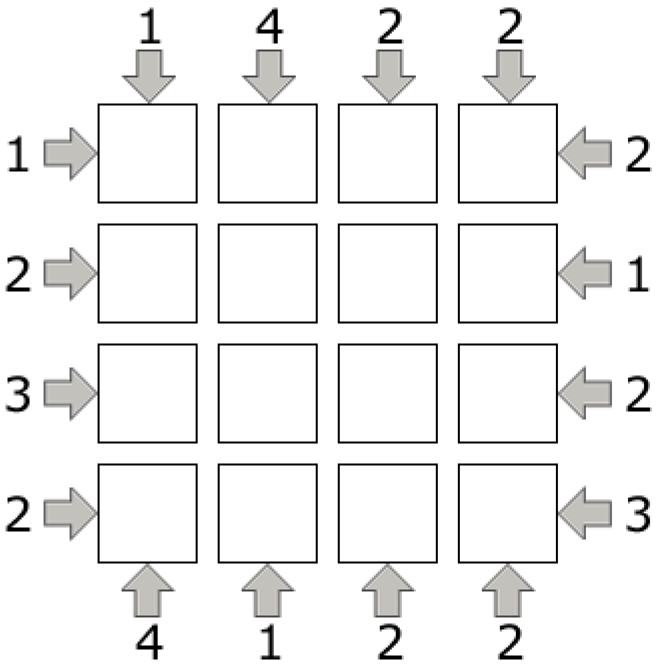


```
(define-fun desk () Int
  5)
(define-fun cupboard () Int
  3)
(define-fun fivedays () Int
  5)
(define-fun blueshirt () Int
  1)
(define-fun Patricia () Int
  2)
(define-fun twentyfivedays () Int
  3)
(define-fun Dana () Int
  3)
(define-fun twentydays () Int
  2)
(define-fun Lori () Int
  4)
(define-fun yellowshirt () Int
  2)
(define-fun fifteendays () Int
  1)
(define-fun thousandtwohundred () Int
  4)
(define-fun fiftyyears () Int
  1)
(define-fun wardrobe () Int
  2)
(define-fun redshirt () Int
  4)
(define-fun thousand () Int
  5)
(define-fun Gina () Int
  5)
(define-fun fiftyfiveyears () Int
  5)
(define-fun Barbara () Int
  1)
(define-fun fortyyears () Int
  3)
(define-fun thousandhundred () Int
  2)
(define-fun greenshirt () Int
  3)
(define-fun ninehundred () Int
  1)
(define-fun eighthundred () Int
  3)
(define-fun orangeshirt () Int
  5)
(define-fun dresser () Int
  4)
(define-fun sixtyyears () Int
  2)
(define-fun table () Int
  1)
(define-fun tendays () Int
  4)
(define-fun fortyfiveyears () Int
  4)
```

Assignment 2:

We began by solving a smaller puzzle first, just to get acquainted with the approach to the problem. Afterwards, we solved the biggest puzzle available.

The small puzzle is as follows:

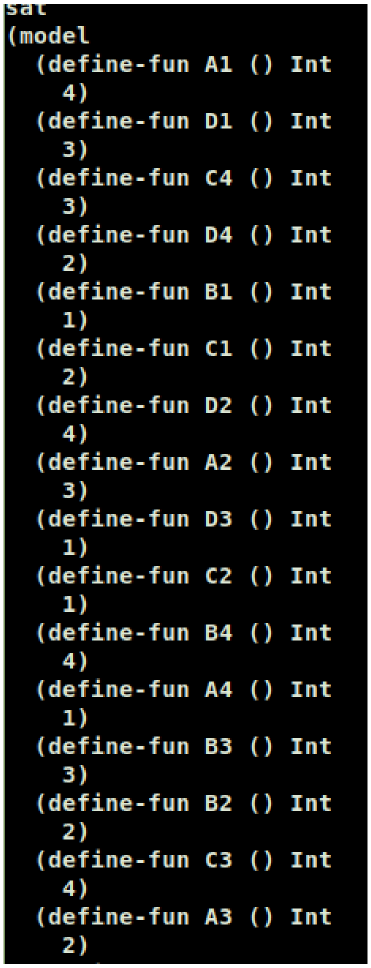


The solution to that puzzle can be found in wk3_2_small.smt2. In that solution, we integer variables to represent the squares, where the name of those variables consists of a letter and a number. The letter represents column and the number – the row (as in MS Excel). We make sure that all the values in each row and column are distinct.

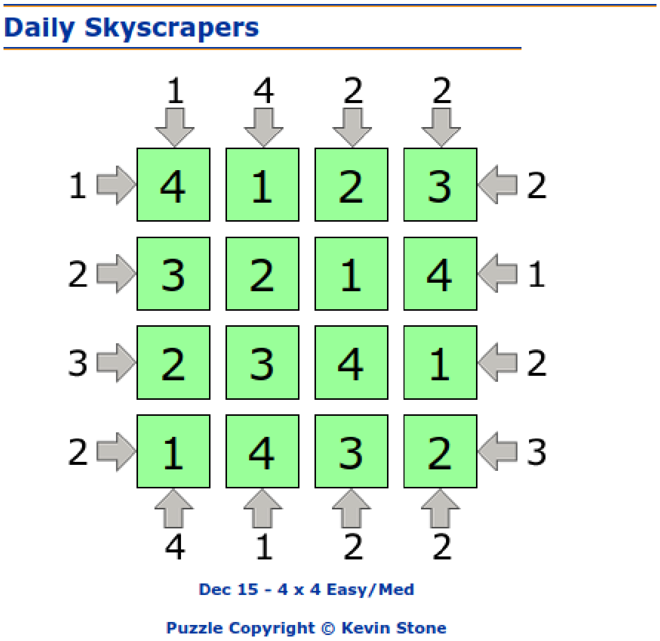
Our approach is to count the number of skyscrapers in each row and column that are bigger than any of the previous skyscrapers in that row or column. That count should equal to the number of the buildings that are seen from the side we started the count from. We repeat that same process from the other side.

We have created functions to help us with that process. The MaxOf functions return the largest integer of their parameters. The Count function counts how many times a larger building than all off the previous ones is seen in a given row or column.

Here is the proof of our solution correctness:



```
sat
(model
  (define-fun A1 () Int
    4)
  (define-fun D1 () Int
    3)
  (define-fun C4 () Int
    3)
  (define-fun D4 () Int
    2)
  (define-fun B1 () Int
    1)
  (define-fun C1 () Int
    2)
  (define-fun D2 () Int
    4)
  (define-fun A2 () Int
    3)
  (define-fun D3 () Int
    1)
  (define-fun C2 () Int
    1)
  (define-fun B4 () Int
    4)
  (define-fun A4 () Int
    1)
  (define-fun B3 () Int
    3)
  (define-fun B2 () Int
    2)
  (define-fun C3 () Int
    4)
  (define-fun A3 () Int
    2)
```



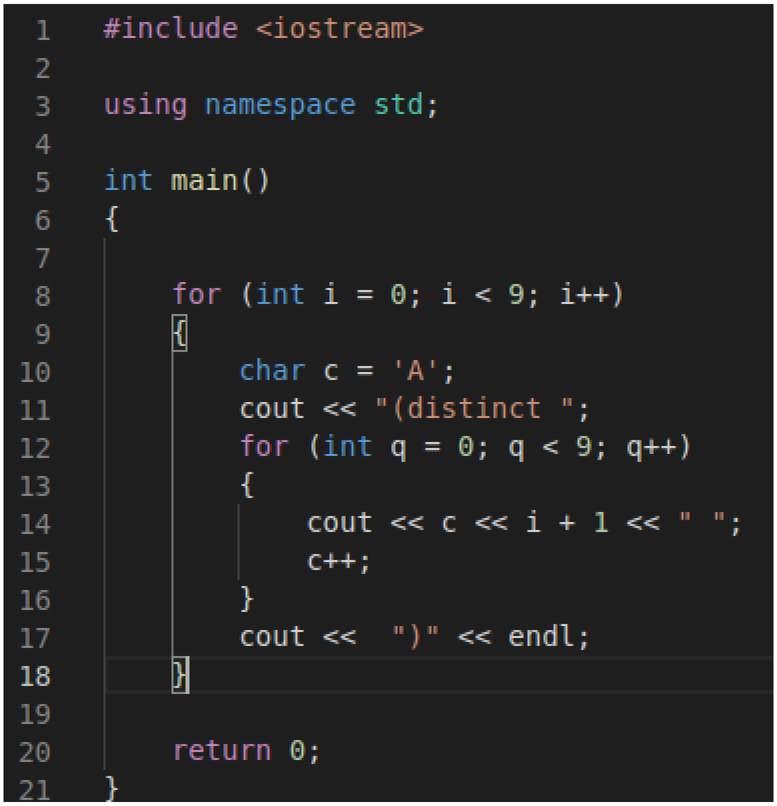
Daily Skyscrapers

	1	4	2	2	
1	4	1	2	3	2
2	3	2	1	4	1
3	2	3	4	1	2
2	1	4	3	2	3
	4	1	2	2	

Dec 15 - 4 x 4 Easy/Med

Puzzle Copyright © Kevin Stone

Now, tackling the 9x9 is done in a very similar way. The solution can be found in wk3_2_large.smt2. We have one extra function, AssertCount, which takes 9 squares of a row or column as parameters, as well as the numbers of buildings seen from each side. It makes the assertion that the number should be as indicated in the puzzle. There are multiple functions named MaxOf, each one handles a different number of parameters, up to 8. We decided we’re not going to type out the repetitive stuff (such as variable declarations and distinct assertions), so we made a small c++ generator, which looks like this:



```cpp
#include <iostream>

using namespace std;

int main()
{

    for (int i = 0; i < 9; i++)
    {
        char c = 'A';
        cout << "(distinct ";
        for (int q = 0; q < 9; q++)
        {
            cout << c << i + 1 << " ";
            c++;
        }
        cout <<  ")" << endl;
    }

    return 0;
}
```

These figures show the puzzle we were solving, as well as the program output and a screenshot with the output in the website:

Daily Skyscrapers

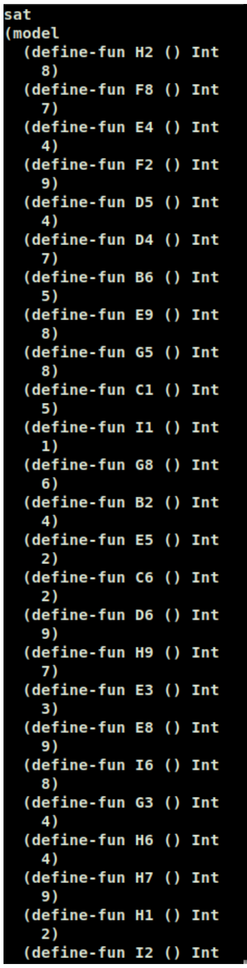
	3	2	4	4	3	2	1	3	3	
3				3						3
4	1						2			3
2		1		5		2		6		1
2							5		2	5
1		7	6			1				4
3					7					2
3		2	4		1					2
3	4		8						5	4
3		6								4
	4	3	1	4	2	4	5	2	5	

Dec 11 - 9 x 9 Easy

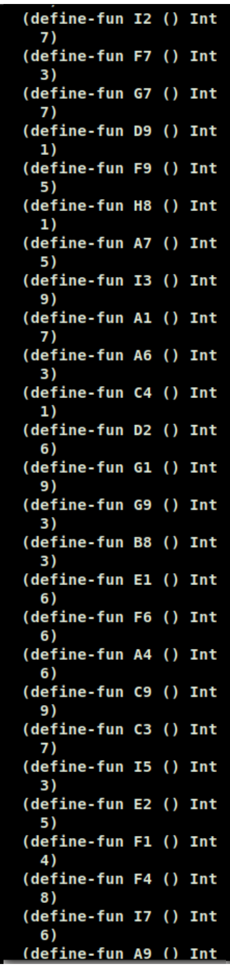
Daily Skyscrapers

	3	2	4	4	3	2	1	3	3	
3	7	8	5	3	6	4	9	2	1	3
4	1	4	3	6	5	9	2	8	7	3
2	8	1	7	5	3	2	4	6	9	1
2	6	9	1	7	4	8	5	3	2	5
1	9	7	6	4	2	1	8	5	3	4
3	3	5	2	9	7	6	1	4	8	2
3	5	2	4	8	1	3	7	9	6	2
3	4	3	8	2	9	7	6	1	5	4
3	2	6	9	1	8	5	3	7	4	4
	4	3	1	4	2	4	5	2	5	

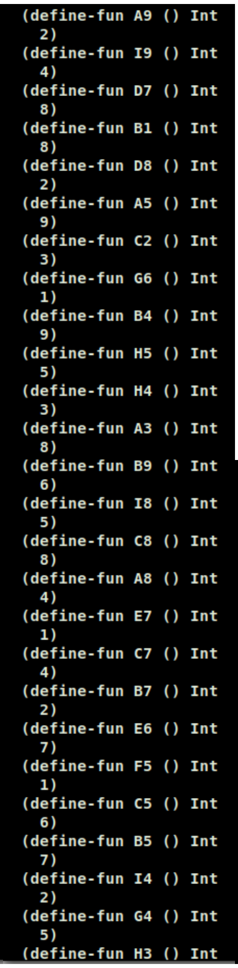
Dec 11 - 9 x 9 Easy



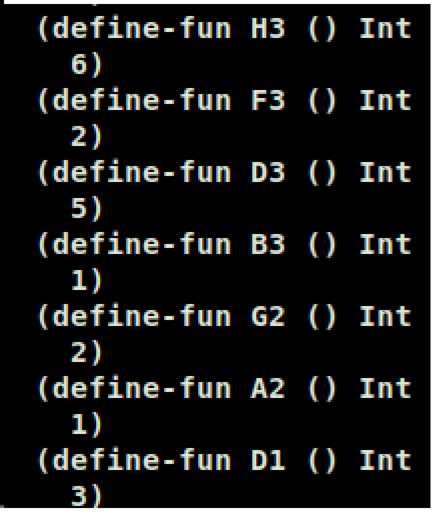
```
sat
(model
  (define-fun H2 () Int
    8)
  (define-fun F8 () Int
    7)
  (define-fun E4 () Int
    4)
  (define-fun F2 () Int
    9)
  (define-fun D5 () Int
    4)
  (define-fun D4 () Int
    7)
  (define-fun B6 () Int
    5)
  (define-fun E9 () Int
    8)
  (define-fun G5 () Int
    8)
  (define-fun C1 () Int
    5)
  (define-fun I1 () Int
    1)
  (define-fun G8 () Int
    6)
  (define-fun B2 () Int
    4)
  (define-fun E5 () Int
    2)
  (define-fun C6 () Int
    2)
  (define-fun D6 () Int
    9)
  (define-fun H9 () Int
    7)
  (define-fun E3 () Int
    3)
  (define-fun E8 () Int
    9)
  (define-fun I6 () Int
    8)
  (define-fun G3 () Int
    4)
  (define-fun H6 () Int
    4)
  (define-fun H7 () Int
    9)
  (define-fun H1 () Int
    2)
  (define-fun I2 () Int
```



```
  (define-fun I2 () Int
    7)
  (define-fun F7 () Int
    3)
  (define-fun G7 () Int
    7)
  (define-fun D9 () Int
    1)
  (define-fun F9 () Int
    5)
  (define-fun H8 () Int
    1)
  (define-fun A7 () Int
    5)
  (define-fun I3 () Int
    9)
  (define-fun A1 () Int
    7)
  (define-fun A6 () Int
    3)
  (define-fun C4 () Int
    1)
  (define-fun D2 () Int
    6)
  (define-fun G1 () Int
    9)
  (define-fun G9 () Int
    3)
  (define-fun B8 () Int
    3)
  (define-fun E1 () Int
    6)
  (define-fun F6 () Int
    6)
  (define-fun A4 () Int
    6)
  (define-fun C9 () Int
    9)
  (define-fun C3 () Int
    7)
  (define-fun I5 () Int
    3)
  (define-fun E2 () Int
    5)
  (define-fun F1 () Int
    4)
  (define-fun F4 () Int
    8)
  (define-fun I7 () Int
    6)
  (define-fun A9 () Int
```



```
  (define-fun A9 () Int
    2)
  (define-fun I9 () Int
    4)
  (define-fun D7 () Int
    8)
  (define-fun B1 () Int
    8)
  (define-fun D8 () Int
    2)
  (define-fun A5 () Int
    9)
  (define-fun C2 () Int
    3)
  (define-fun G6 () Int
    1)
  (define-fun B4 () Int
    9)
  (define-fun H5 () Int
    5)
  (define-fun H4 () Int
    3)
  (define-fun A3 () Int
    8)
  (define-fun B9 () Int
    6)
  (define-fun I8 () Int
    5)
  (define-fun C8 () Int
    8)
  (define-fun A8 () Int
    4)
  (define-fun E7 () Int
    1)
  (define-fun C7 () Int
    4)
  (define-fun B7 () Int
    2)
  (define-fun E6 () Int
    7)
  (define-fun F5 () Int
    1)
  (define-fun C5 () Int
    6)
  (define-fun B5 () Int
    7)
  (define-fun I4 () Int
    2)
  (define-fun G4 () Int
    5)
  (define-fun H3 () Int
```



```
  (define-fun H3 () Int
    6)
  (define-fun F3 () Int
    2)
  (define-fun D3 () Int
    5)
  (define-fun B3 () Int
    1)
  (define-fun G2 () Int
    2)
  (define-fun A2 () Int
    1)
  (define-fun D1 () Int
    3)
```