

Testfaelle zu Aufgabenblatt 9

-- Testfaelle fuer Aufgabenteil 1

-- a)

```
isValidSL [[0,5,4,3,2,1,0],[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],
[3,30,40,50,10,20,2],[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[0,1,2,2,2,2,0]] ==
True
isValidSL [[1,5,4,3,2,1,0],[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],
[3,30,40,50,10,20,2],[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[0,1,2,2,2,2,0]] ==
True
isValidSL [[0,6,4,3,2,1,0],[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],
[3,30,40,50,10,20,2],[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[0,1,2,2,2,2,0]] ==
False
isValidSL [[0,5,4,3,2,1,0],[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],
[3,30,40,50,20,20,2],[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[0,1,2,2,2,2,0]] ==
False
isValidSL [[0,5,4,3,2,1,0],[5,10,20,30,40,50,1],[4,20,30,50,10,2],
[3,30,40,50,10,20,2],[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[0,1,2,2,2,2,0]] ==
False
isValidSL [] == False
```

-- b)

-- following helper function compares two skylines

```
compareSkyLine s1 s2 =
  zipWith id (conv s1) s1 == zipWith id (conv s2) s2
  where conv m = [init . tail] ++ (replicate (length m - 2) id) ++ [init . tail]

compareSkyLine (compVisibility [[10,20,30,40,50],[20,30,40,50,10],
[30,40,50,10,20],[40,50,10,20,30],[50,10,20,30,40]]) ([0,5,4,3,2,1,0],
[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],[3,30,40,50,10,20,2],
[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[0,1,2,2,2,2,0]]) == True
compareSkyLine (compVisibility [[10,20,30,40,50],[20,30,40,50,10],
[30,40,50,10,20],[40,50,10,20,30],[50,10,20,30,40]]) ([1,5,4,3,2,1,2],
[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],[3,30,40,50,10,20,2],
[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[3,1,2,2,2,2,4]]) == True
compareSkyLine (compVisibility [[10,20,30,40,50],[20,30,40,50,10],
[30,40,50,10,20],[40,50,10,20,30],[50,10,20,30,40]]) ([1,5,4,3,2,1,2],
[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],[3,30,40,50,10,20,2],
[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[3,1,2,2,3,2,4]]) == False
compareSkyLine (compVisibility [[10,20,30,40,50],[20,30,40,50,10],
[30,40,50,10,20],[40,50,10,20,30],[50,10,20,30,40]]) ([1,5,4,3,2,1,2],
[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],[3,30,40,50,10,20,2],
[2,40,1337,10,20,30,2],[1,50,10,20,30,40,2],[3,1,2,2,2,2,4]]) == False
compareSkyLine (compVisibility [[10,20,30,40,50],[20,30,40,50,10],
[30,40,50,10,20],[40,50,10,20,30],[50,10,20,30,40]]) ([5,10,20,30,40,50,1],
[4,20,30,40,50,10,2],[3,31,40,50,10,20,2],[2,40,50,10,20,30,2],
[1,50,10,20,30,40,2]]) == False
```

-- c)

-- do note: proposed solutions might not be unique!

```
buildSkyscrapers [[0,2,1,0],[2,0,0,1],[1,0,0,2],[0,1,2,0]] == Just [[0,2,1,0],
[2,10,20,1],[1,20,10,2],[0,1,2,0]]
buildSkyscrapers [[0,2,2,0],[2,0,0,1],[1,0,0,2],[0,1,2,0]] == Nothing
buildSkyscrapers [[0,5,4,3,2,1,0],[5,0,0,0,0,0,1],[4,0,0,0,0,0,2],[3,0,0,0,0,0,2],
```

```

[2,0,0,0,0,0,2],[1,0,0,0,0,0,2],[0,1,2,2,2,2,0]] == Just [[0,5,4,3,2,1,0],
[5,10,20,30,40,50,1],[4,20,30,40,50,10,2],[3,30,40,50,10,20,2],
[2,40,50,10,20,30,2],[1,50,10,20,30,40,2],[0,1,2,2,2,2,0]]
buildSkyscrapers [[0,6,4,3,2,1,0],[5,0,0,0,0,0,1],[4,0,0,0,0,0,2],[3,0,0,0,0,0,2],
[2,0,0,0,0,0,2],[1,0,0,0,0,0,2],[0,1,2,2,2,2,0]] == Nothing
buildSkyscrapers [[0,5,4,3,2,1,0],[5,0,0,0,0,0,1],[4,0,0,0,0,0,2],[3,0,0,0,0,0,2],
[2,0,0,0,0,0,2],[1,0,0,0,0,0,2],[0,1,2,2,2,2,1,0]] == Nothing
buildSkyscrapers [[0,5,1,3,2,4,0],[2,0,0,0,0,0,3],[3,0,0,0,0,0,2],[2,0,0,0,0,0,2],
[2,0,0,0,0,0,1],[1,0,0,0,0,0,4],[0,1,2,2,2,2,0]] == Just [[0,5,1,3,2,4,0],
[2,10,50,30,40,20,3],[3,20,10,40,50,30,2],[2,30,20,50,10,40,2],
[2,40,30,10,20,50,1],[1,50,40,20,30,10,4],[0,1,2,2,2,2,0]]

-- Weitere Testfaelle fuer Aufgabenteil 1

skyline1_ok = [[0,3,1,2,2,3,0],[2,30,50,40,20,10,4],[3,10,30,20,50,40,2],
[2,20,10,50,40,30,3],[2,40,20,30,10,50,1],[1,50,40,10,30,20,4],[0,1,2,3,3,2,0]]
skyline1_ok_start = [[0,3,1,2,2,3,0],[2,0,0,0,0,0,4],[3,0,0,0,0,0,2],
[2,0,0,0,0,0,3],[2,0,0,0,0,0,1],[1,0,0,0,0,0,4],[0,1,2,3,3,2,0]]
skyline1_ok_novis = [[30,50,40,20,10],[10,30,20,50,40],[20,10,50,40,30],
[40,20,30,10,50],[50,40,10,30,20]]

skyline2_ok = [[0,2,2,3,1,3,0],[2,40,20,10,50,30,2],[1,50,10,40,30,20,4],
[2,30,50,20,40,10,3],[3,10,30,50,20,40,2],[3,20,40,30,10,50,1],[0,3,2,2,4,1,0]]
skyline2_ok_start = [[0,2,2,3,1,3,0],[2,0,0,0,0,0,2],[1,0,0,0,0,0,4],
[2,0,0,0,0,0,3],[3,0,0,0,0,0,2],[3,0,0,0,0,0,1],[0,3,2,2,4,1,0]]
skyline2_ok_novis = [[40,20,10,50,30],[50,10,40,30,20],[30,50,20,40,10],
[10,30,50,20,40],[20,40,30,10,50]]

skyline3_nok_start = [[0,3,1,2,2,3,0],[2,0,0,0,0,0,4],[3,0,0,0,0,0,2],
[2,0,0,0,0,0,3],[2,0,0,0,0,0,2],[1,0,0,0,0,0,4],[0,1,2,3,3,2,0]]
skyline4_nok_start = [[0,2,2,3,1,3,0],[2,0,0,0,0,0,2],[1,0,0,0,0,0,4],
[2,0,0,0,0,0,3],[3,0,0,0,0,0,2],[3,0,0,0,0,0,1],[0,3,2,2,4,2,0]]
skyline5_nok_start = [[0,2,2,3,1,3,0],[2,0,0,0,0,0,2],[6,0,0,0,0,0,4],
[2,0,0,0,0,0,3],[3,0,0,0,0,0,2],[3,0,0,0,0,0,1],[0,3,2,2,4,2,0]]

skyline6_nok = [[0,2,2,3,1,3,0],[2,40,20,10,50,30,3],[1,50,10,40,30,20,4],
[2,30,50,20,40,10,3],[3,10,30,50,20,40,2],[3,20,40,30,10,50,1],[0,3,2,2,4,1,0]]
skyline7_nok = [[0,2,2,3,1,3,0],[2,40,20,10,50,30,2],[1,50,10,40,30,20,4],
[2,30,50,20,40,10,6],[3,10,30,50,20,40,2],[3,20,40,30,10,50,1],[0,3,2,2,4,1,0]]
skyline8_nok = [[0,2,2,3,1,3,0],[2,40,20,10,50,30,2],[1,50,10,40,30,20,4],
[2,30,50,20,40,10,3],[3,10,30,50,20,60,2],[3,20,40,30,10,50,1],[0,3,2,2,4,1,0]]
skyline9_nok = [[0,2,2,3,1,3,0],[2,40,20,10,0,30,2],[1,50,10,40,30,20,4],
[2,30,50,20,40,10,3],[3,10,30,50,20,-10,2],[3,20,40,30,10,50,1],[0,3,2,2,4,1,0]]

-- a)
testsV = [skyline1_ok, skyline2_ok, skyline6_nok, skyline7_nok, skyline8_nok,
skyline9_nok]

testValid = (map (isValidSL) testsV) == [True, True, False, False, False, False]

-- b)
-- following helper function compares two skylines (from hassi)
compareSkyLine s1 s2 = zipWith id (conv s1) s1 == zipWith id (conv s2) s2
                        where conv m = [init . tail] ++ (replicate
(length m - 2) id) ++ [init . tail]

testsC = [(compareSkyLine (compVisibility skyline1_ok_novis) (skyline1_ok)),
(compareSkyLine (compVisibility skyline2_ok_novis) (skyline2_ok))]
testComp = (foldl1 (&) True (testsC))

```

```

-- c)
-- solutions inconclusively unique

testsB = [skyline1_ok_start, skyline2_ok_start, skyline3_nok_start,
skyline4_nok_start, skyline5_nok_start]
solutionsB = [(Just skyline1_ok), (Just skyline2_ok), Nothing, Nothing, Nothing]

testBuild = (map (buildSkyscrapers) testsB) == solutionsB

testallSkylines = testValidSL && testComp && testBuild


-- Testfaelle zu Aufgabenteil 2
-- basic start sudokus

basic_start1 = [[9,1,6,0,0,4,0,7,2],
                [8,0,0,6,2,0,0,5,0],
                [5,0,0,0,0,8,9,3,0],
                [0,6,0,0,0,0,2,0,0],
                [0,0,0,2,0,7,0,0,0],
                [0,0,5,0,0,0,0,9,0],
                [0,9,7,8,0,0,0,0,3],
                [0,8,0,0,7,6,0,0,9],
                [4,5,0,1,0,0,6,8,7]]

basic_start2 = [[6,0,0,3,0,0,1,0,0],
                [0,7,1,6,2,0,0,0,0],
                [8,0,5,0,0,1,0,0,0],
                [5,0,0,8,7,0,9,0,1],
                [0,0,9,0,0,0,6,0,0],
                [4,0,7,0,6,9,0,0,8],
                [0,0,0,2,0,0,8,0,7],
                [0,0,0,0,8,6,4,1,0],
                [0,0,8,0,0,3,0,0,2]]

basic_start3 = [[9,0,6,0,1,3,0,0,8],
                [0,5,8,0,0,0,0,9,0],
                [0,3,0,0,0,0,0,1,0],
                [0,6,0,8,0,0,9,2,0],
                [0,0,3,4,0,9,1,0,0],
                [0,4,9,0,0,6,0,3,0],
                [0,9,0,0,0,0,0,8,0],
                [0,1,0,0,0,0,6,7,0],
                [4,0,0,9,6,0,3,0,1]]

basic_inv1 = [[9,0,6,0,1,3,0,0,8],
               [0,5,8,0,0,0,0,9,0],
               [0,3,0,0,0,0,0,1,0],
               [0,6,0,8,0,0,9,2,0],
               [0,0,3,4,0,9,9,0,0],
               [0,4,9,0,0,6,0,3,0],
               [0,9,0,0,0,0,0,8,0],
               [0,1,0,0,0,0,6,7,0],
               [4,0,0,9,6,0,3,0,1]]

```

```

basic_inv2 = [[9,0,6,0,1,3,0,0,8],
              [1,5,8,0,0,0,0,9,0],
              [0,3,1,0,0,0,0,1,0],
              [0,6,0,8,0,0,9,2,0],
              [0,0,3,4,0,9,1,0,0],
              [0,4,9,0,0,6,0,3,0],
              [0,9,0,0,0,0,0,8,0],
              [0,1,0,0,0,0,6,7,0],
              [4,0,0,9,6,0,3,0,1]]

-- cross start sudokus

cross_start1 = [[0,0,3,4,0,0,0,8,0],
                [4,5,6,7,8,9,1,2,0],
                [7,8,0,1,2,3,0,5,0],
                [0,6,4,2,1,7,9,3,0],
                [2,9,1,5,3,8,0,7,4],
                [3,0,8,6,9,4,0,0,0],
                [0,3,5,9,0,2,0,4,1],
                [0,1,7,8,0,0,0,6,2],
                [0,0,2,0,6,1,0,0,0]]

cross_start2 = [[0,0,3,4,0,0,0,0,0],
                [4,5,0,7,8,9,1,0,0],
                [7,8,0,1,2,3,0,5,0],
                [0,6,4,2,1,7,9,0,0],
                [2,9,1,5,0,0,0,7,4],
                [0,0,8,6,9,4,0,0,0],
                [0,0,5,9,0,2,0,4,1],
                [0,0,7,8,0,0,0,6,2],
                [0,0,2,0,0,1,0,0,0]]

cross_inv1 = [[0,0,3,4,0,0,0,0,0],
              [4,5,0,7,8,9,1,0,0],
              [7,8,0,1,2,3,0,5,0],
              [0,6,4,2,1,7,9,0,0],
              [2,9,1,5,0,0,0,7,4],
              [0,0,8,6,9,4,0,0,0],
              [0,0,5,9,0,2,0,4,1],
              [0,0,7,8,0,0,0,6,2],
              [0,0,2,0,0,1,0,0,6]]

-- color start sudokus

color_start1 = [[1,2,3,0,5,0,0,0,9],
                [0,0,0,0,0,0,0,0,3],
                [7,0,9,0,2,3,4,5,6],
                [2,0,1,0,0,4,8,9,5],
                [8,0,5,9,1,2,0,0,0],
                [0,0,0,5,3,8,2,7,1],
                [0,0,0,3,6,7,5,4,8],
                [5,4,8,2,9,0,0,0,0],
                [0,6,7,8,4,0,0,0,0]]

color_start2 = [[0,0,3,4,5,0,0,0,0],
                [0,0,6,0,0,0,1,2,3],
                [0,8,9,1,2,3,4,5,6],
                [0,3,1,6,7,4,8,9,5],
                [0,7,5,0,0,0,3,0,0],

```

```

        [6,9,4,5,3,0,2,0,0],
        [9,1,2,3,0,0,0,0,0],
        [0,0,0,2,9,0,6,0,0],
        [0,0,0,8,4,0,0,0,0]]

color_inv1  = [[0,0,3,4,5,0,0,0,0],
               [0,0,6,0,0,6,1,2,3],
               [0,8,9,1,2,3,4,5,6],
               [0,3,1,6,7,4,8,9,5],
               [0,7,5,0,0,0,3,0,0],
               [6,9,4,5,3,0,2,0,0],
               [9,1,2,3,0,0,0,0,0],
               [0,0,0,2,9,0,6,0,0],
               [0,0,0,8,4,0,0,0,0]]

-- 2.a

t_val1 = (isValidSDK basic_start1 Basic) == True
t_val2 = (isValidSDK basic_start2 Basic) == True
t_val3 = (isValidSDK basic_start3 Basic) == True
t_val4 = (isValidSDK cross_start1 Cross) == True
t_val5 = (isValidSDK cross_start2 Cross) == True
t_val6 = (isValidSDK color_start1 Color) == True
t_val7 = (isValidSDK color_start2 Color) == True
t_val8 = (isValidSDK basic_inv1 Basic) == False
t_val9 = (isValidSDK basic_inv2 Basic) == False
t_val10 = (isValidSDK cross_inv1 Cross) == False
t_val11 = (isValidSDK color_inv1 Color) == False

test_isValid = and [t_val1,t_val2,t_val3,t_val4,t_val5,t_val6,t_val7,
                    t_val8,t_val9,t_val10,t_val11]

-- 2.b
-- * Beachten Sie, dass die voll ausgefuellte Sudoku zu einer Anfangs-Sudoku-
Matrix i.a nicht eindeutig ist

-- Basis-Instanzen

t_basic1 = (solve basic_start1 Basic) == (Just [[9,1,6,3,5,4,8,7,2],
[8,7,3,6,2,9,1,5,4],[5,2,4,7,1,8,9,3,6],[7,6,8,9,3,5,2,4,1],[1,4,9,2,8,7,3,6,5],
[2,3,5,4,6,1,7,9,8],[6,9,7,8,4,2,5,1,3],[3,8,1,5,7,6,4,2,9],[4,5,2,1,9,3,6,8,7]])
t_basic2 = (solve basic_start2 Basic) == (Just [[6,2,4,3,5,7,1,8,9],
[9,7,1,6,2,8,3,5,4],[8,3,5,4,9,1,7,2,6],[5,6,3,8,7,2,9,4,1],[2,8,9,1,3,4,6,7,5],
[4,1,7,5,6,9,2,3,8],[3,4,6,2,1,5,8,9,7],[7,5,2,9,8,6,4,1,3],[1,9,8,7,4,3,5,6,2]])
t_basic3 = (solve basic_start3 Basic) == (Just [[9,7,6,5,1,3,2,4,8],
[1,5,8,6,4,2,7,9,3],[2,3,4,7,9,8,5,1,6],[7,6,1,8,3,5,9,2,4],[8,2,3,4,7,9,1,6,5],
[5,4,9,1,2,6,8,3,7],[6,9,7,3,5,1,4,8,2],[3,1,5,2,8,4,6,7,9],[4,8,2,9,6,7,3,5,1]])
t_basic4 = (solve basic_inv1 Basic) == Nothing
t_basic5 = (solve basic_inv2 Basic) == Nothing

test_basic = and [t_basic1, t_basic2, t_basic3, t_basic4, t_basic5]

-- Kreuz-Instanzen

t_cross1 = (solve cross_start1 Cross) == (Just [[1,2,3,4,5,6,7,8,9],
[4,5,6,7,8,9,1,2,3],[7,8,9,1,2,3,4,5,6],[5,6,4,2,1,7,9,3,8],[2,9,1,5,3,8,6,7,4],
[3,7,8,6,9,4,2,1,5],[6,3,5,9,7,2,8,4,1],[9,1,7,8,4,5,3,6,2],[8,4,2,3,6,1,5,9,7]])

```

```

t_cross2 = (solve cross_start2 Cross) == (Just [[1,2,3,4,5,6,7,8,9],
[4,5,6,7,8,9,1,2,3],[7,8,9,1,2,3,4,5,6],[5,6,4,2,1,7,9,3,8],[2,9,1,5,3,8,6,7,4],
[3,7,8,6,9,4,2,1,5],[6,3,5,9,7,2,8,4,1],[9,1,7,8,4,5,3,6,2],[8,4,2,3,6,1,5,9,7]])
t_cross3 = (solve cross_inv1 Cross) == Nothing

test_cross = and [t_cross1, t_cross2, t_cross3]

-- Farb-Instanzen
-- Die Loesungen zu Color Instanzen sind nicht eindeutig

t_color1 = (solve color_start1 Color) == (Just [[1,2,3,4,5,6,7,8,9],
[4,5,6,7,8,9,1,2,3],[7,8,9,1,2,3,4,5,6],[2,3,1,6,7,4,8,9,5],[8,7,5,9,1,2,3,6,4],
[6,9,4,5,3,8,2,7,1],[9,1,2,3,6,7,5,4,8],[5,4,8,2,9,1,6,3,7],[3,6,7,8,4,5,9,1,2]])
t_color2 = (solve color_start2 Color) == (Just [[1,2,3,4,5,6,7,8,9],
[4,5,6,7,8,9,1,2,3],[7,8,9,1,2,3,4,5,6],[2,3,1,6,7,4,8,9,5],[8,7,5,9,1,2,3,6,4],
[6,9,4,5,3,8,2,7,1],[9,1,2,3,6,7,5,4,8],[5,4,8,2,9,1,6,3,7],[3,6,7,8,4,5,9,1,2]])
t_color3 = (solve color_inv1 Color) == Nothing

test_color = and [t_color1, t_color2, t_color3]

test_all = and [test_isValid, test_basic, test_cross, test_color]

```