

Photographing Skyscrapers (Racket+Haskell - 7+7 Points)

May 25, 2023

You are an avid photographer that is obsessed with regular structures and you want to take pictures of cities that are built on regular grids. It turns out that you are also really into roof tops so you want to see as many of them in your pictures as possible. Naturally, you wonder from which side of a given city (North/South/East/West) you should be taking the picture. Luckily a befriended architect gave you maps of the cities you want to photograph. The maps are very simplified and can be represented as lists of lists of integers, so for example in Scheme:

```
;    north
(define city
  '((3 0 3 7 3)
    (2 5 5 1 2)
    (6 5 3 3 2)
    (3 3 5 4 9)
    (3 5 3 9 0)))
;    south
```

Every number represents the height of a building.

A roof is *visible* if all other roofs between it and the edge of the grid are *smaller* than it. Only consider roofs in the same row or column. The first roof at the edge of a grid is always visible.

1 Task 3 - Racket

In **Racket**, write a function (**best-view city**) that outputs the direction with the most roofs visible, along with the number of roofs visible from that direction. The direction should be one of four symbols: 'N, 'S, 'E, or 'W. The result should be a pair in the format '(**direction** . **number**).

```
(define city
  '((3 3 3)
    (1 2 3)
    (1 2 3)))

; 'N has 3 roofs, 'S has 5, 'E has 3, and 'W is the best with 7
> (best-view city)
'(W . 7)
```

Your file should be called `task3.rkt` and should export the `best-view` function.

```
#lang racket
(provide best-view)
(define (best-view city)
  ; Implement me!
)
```

2 Task 4 - Haskell

In Haskell, write a function `bestView` `city` that outputs the direction with the most roofs visible, along with the number of roofs visible from that direction. The direction should be one of four characters: `'N'`, `'S'`, `'E'`, or `'W'`. The result should be a pair in the format `(direction, number)`.

```
city = [[3, 3, 3],
        [1, 2, 3],
        [1, 2, 3]]

-- 'N' has 3 roofs, 'S' has 5, 'E' has 3, and 'W' is the best with 7
bestView city -- ('W', 7)
```

Your file should be called `Task4.hs` and should export the `bestView` function.

```
module Task4 (bestView) where

bestView :: [[Int]] -> (Char, Int)
bestView city = (' ', 0) -- Implement me!
```