CS 5035 (Fall 2016)

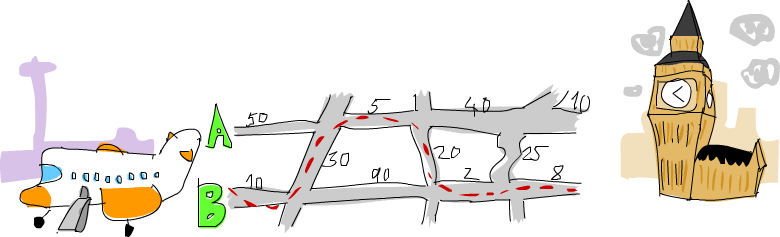
### Project 9. Functionally Solving Problems

Based on chapter [10 of LYH](http://learnyouahaskell.com/functionally-solving-problems).  **(no videos)**

#### **[Heathrow to London calculator](http://learnyouahaskell.com/functionally-solving-problems" \l "heathrow-to-london) (first attempt by Nov 21).**

#### **Your plane has just landed in England and you rent a car. You have a meeting really soon and you have to get from Heathrow Airport to London as fast as you can (but safely!).**

#### **There are two main roads going from Heathrow to London and there's a number of regional roads crossing them. It takes you a fixed amount of time to travel from one crossroads to another. It's up to you to find the optimal path to take so that you get to London as fast as you can! You start on either side.**



See the chapter for the code. In your version you can assume the input is given as a list and isn’t read from the terminal.

My solution is available [here](https://drive.google.com/file/d/0B-I58s-_d3o5eVZRXzdlRzBwVGs/view?usp=sharing).

As the text noted, repeatedly summing the distances in a Path is inefficient if the paths get long. So let’s add a distance field to the Path data type. To help you develop a feeling for pattern matching and whether you like it, revise the Path data type in two ways.

Unnamed fields: **data** Path = Path [Step] Int

Named fields: **data** Path = Path {steps :: [Step], dist :: Int}

Then revise the rest of the code so that it works with the modified Path definitions. When you use named fields, don’t use pattern matching on Paths elsewhere in the code. Use the field names as getters and setters instead.

Field names are used as setters when you create a new instance of a type. For example, the +> operation might be written as follows when using named fields. (Note that pattern matching works even when you have named fields. But for this project, don’t use pattern matching when you have named fields.)

(+>) :: Path -> Path -> Path

p1 +> p2 = Path {steps = steps p1 ++ steps p2,

dist = dist p1 + dist p2}

When you are done, you will have two revised files for this project. They should be the same except for how Paths are treated.