

CMSC 15100 Introduction to Computer Science I

Homework 3

Important: Please add to (require "../include/cs151-core.rkt") at top of your file. Your answers should be saved in a single .rkt file and submitted via Canvas before the start of the next lecture. At the top of the file for this assignment and all future assignments include a comment,

```
;; Homework 3
;;
;; your name
;; your CNET id
```

If your code doesn't compile, it may not be graded. Please double check to make sure your code runs before submitting!

Problems

1. (15 points) For this problem, you should copy the definition of the `Point` struct from class into your .rkt file.

When you include the hospital, the Logan Center, the Law School, and Stagg Field, UChicago has a pretty big campus. We can make the campus into a coordinate grid by putting the origin at the center of the quad, and by agreeing that the positive x direction is east, and that the positive y direction is north. At the top of the next page is a map of the campus overlaid with a coordinate grid (a larger copy of this grid can be found on Canvas).

In this grid, the origin is at the center of the main quad, and each square has side length one unit. So, for example, on this map, Ryerson Hall is approximately at coordinates $(0.25, 0.4)$. Now that we have a coordinate grid set up, we can represent places around UChicago in Racket by `Points`.

You don't have to write any tests for this entire problem. You should test your code in the interaction window and check that the output makes sense.

- (a) Define at least 4 variables with type `Point` for different places around campus, including a variable `ryerson` with the location of Ryerson Hall. You don't have to squint too hard at the locations, just pick some reasonable coordinates.

When you want to check that your functions work in the other parts, use these points in the interaction window and check that the output you get is reasonable. For example,

```
(dist ryerson ratner)
```

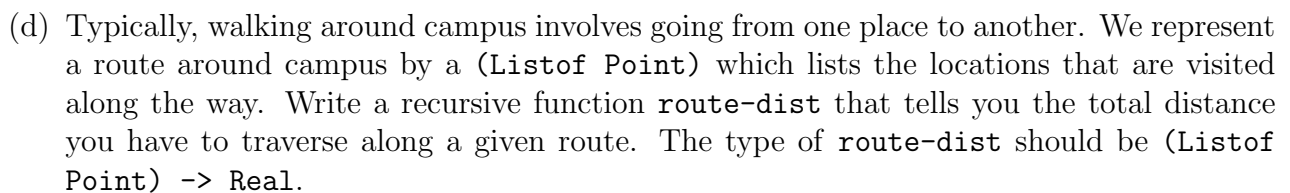
should give you something close to 3.

- (b) On this map, the quad is approximately the rectangular region between the four corners $(0.75, -1)$, $(-0.75, -1)$, $(0.75, 1)$ and $(-0.75, 1)$.

Write a function `in-quad?` that checks if a `Point` is in the quad.

- (c) Write a function `dist` that computes the straight-line distance between two `Points`. For points (x_1, y_1) and (x_2, y_2) this distance is given by

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$



Hint: just like the function `first`, there is a function `second`, which takes a list and gives you the second element of the list.

- ```
(define-type Color (U 'gray 'red 'white 'blue))
(define-type Size (U 'small 'medium 'large))

(define-struct Shirt
 ([short-sleeves? : Boolean]
```

```
[color : Color]
[size : Size]))
(define-struct Pants
 ([shorts? : Boolean]
 [color : Color]
 [size : Size]))
```

```
(define-type Clothing (U Shirt Pants))
```

- (a) Make a function `clothing-color` that takes a piece of clothing and outputs the color. The type should be `Clothing -> Color`.
- (b) Make a function `same-size?` that takes two clothing items, and determines if they have the same size. The type is `Clothing Clothing -> Boolean`.
- (c) When it gets to be really hot outside, a short-sleeve shirt can help keep cool, and so can shorts. Implement a function `cool?` that takes an input of type `Clothing` and returns a `Boolean` if it's a short-sleeve shirt, or if it's shorts.
- (d) (1 point extra credit) Make a function `draw-clothing` that uses the image library from Lab 1 to create a reasonable picture of the clothing item. Make sure that the picture accurately reflects the traits of the item!