E-402-STFO PROBLEMS FOR MODULE 6

CREATED BY HENNING ULFARSSON

This module is concerned with dropping needles to approximate π and a robotic arm.

You get a perfect score for this module by getting 40 points or more.

1. Dropping needles (15 points)

Problem 1 (5 points)**.** Imagine we have a needle of length 1 and an infinite ruled paper with parallel horizontal lines distanced 1 apart. Write a function `m6p1(n)` that given a positive integer `n` simulates dropping of the needle onto the paper by choosing a point on the paper at random by first choosing a point (x, y) for the tip of the needle and then a random angle from the positive x-axis to the other end of the needle. (You may choose the point (x, y) to lie inside some large rectangle if it helps.) The output of the function should be the fraction of times the needle intersects one of the lines.

```
Input: n = 7
Run: m6p1(n)
Output: 4/7
```

Problem 2 (10 points)**.** By choosing n large in the above problem approximates $2/\pi$. Why? (Turn in a LaTeX-ed PDF).

2. Robotics (45 points)

This section assumes you have read through the file Robotics.sws which came with the homework.

In all the problems in this section we are working with a robotic arm made of two rods. The first has length 1 and is mounted at the point $(0, 0)$ and can rotate in a full circle around that point. At the other end of this rod we have another rod of length 1 mounted, and that second rod can also rotate in a full circle (althoug it *can not move through* the first rod). At the end of the second rod we have a *hand* or some other equipment capable of grabbing stuff.

Problem 3 (5 points)**.** Write a function `m6p3(x,y)` that given a point `(x,y)` returns `True` if the arm can reach the point `(x,y)`, and `False` otherwise.

```
Input: x = 1, y = 0.5
Run: m6p3(x,y)
Output: True
```

Problem 4 (25 points)**.** Write a function `m6p4(x,y,z,w,n)` that given points `(x,y)` and `(z,w)` reachable by the arm returns a *valid* list of `n` coordinates that the arm can travel through to go from `(x,y)` to `(z,w)`. Each entry in the list should be of the form `(a,b,c,d)` where `(a,b)` is the point representing the location of the joint and `(c,d)` is the point representing the location of the hand. The first entry in the list should be `x',y',x,y`, i.e, the initial position, and the last entry in the

Date: Updated November 8, 2013.

list should be the `z',w',z,w`, i.e., the ending position. We also require that for two adjacent entries in the list, `a,b,c,d` and `a',b',c',d'`, that the distance between the points `(a,b)` and `(a',b')` is at most the Euclidean distance between `(x,y)` and `(z,w)` multiplied by `10/(n-1)`. We place the same requirement on `(c,d)` and `(c',d')`.

```
Input: x = 1, y = 0.5, z = -0.6, w = 0.9, n =  5
Run: L = m6p4(x,y,z,w,n)
     for a in L:
         print a
Output: (-1/20*sqrt(55) + 1/2, 1/10*sqrt(11)*sqrt(5) + 1/4, 1, 0.500000000000000)
        (-1/10*sqrt(41) + 3/10, 1/10*sqrt(41) + 3/10, 3/5, 0.600000000000000)
        (-7/1060*sqrt(18391) + 1/10, 1/530*sqrt(347)*sqrt(53) + 7/20, 1/5, 0.7000000000
        (-4/9996896733045*sqrt(4879323162451065227097849) - 1/10, -1/9996896733045*sqrt
        (-3/260*sqrt(3679) - 3/10, -1/130*sqrt(283)*sqrt(13) + 9/20, -3/5, 0.9000000000
```

Problem 5 (15 points)**.** Create a Sage worksheet (`.sws`) (or a plain `.sage` file) that animates a robotic arm as above picking up randomly placed points and collecting them in one place.

School of Computer Science, Reykjavik University, Menntavegi 1, 101 Reykjavík, Iceland
E-mail address: `henningu@ru.is`