

---

## MATH 20 FALL 14 ASSIGNMENT 3, DUE FRIDAY 10/3

You're encouraged to discuss these problems with other students in the class. Hand in the solutions to the book problems and part 2 on paper at the beginning of class on Friday, 10/3. Send the code for part 3 by email to [ewa.j.infeld.gr@dartmouth.edu](mailto:ewa.j.infeld.gr@dartmouth.edu) by the same time.

### 1 Book problems

Section 3.1: Problems 16, 17

Section 3.2: Problems 8, 22, 23, 33

### 2 Middle point distribution

Make 3 marks on a unit stick, at points chosen uniformly at random. What is the probability density for the position of the middle mark (i.e. the one with position between the two other marks)?

Hint: this is no longer uniform.

### 3 Fixed points

Write code that generates a permutation of  $\{1, 2, \dots, 10\}$  uniformly at random, term by term, and then prints out a list of fixed points in that permutation.

#### 3.1 Some useful commands

You can have a **for** loop go directly through entries in a list. If  $P$  is a permutation in form of a list, then:

```
for i in P:
    (instructions)
```

executes the instructions with  $i$  being each entry in the list in turn, i.e.  $i=P[0]$ ,  $i=P[1]$ ,  $\dots$ . However, you might find that going through  $i$  as a position in the list serves your code better.

Another thing that may prove very useful for this task is a **while** loop. Like an **if** statement, it has a condition and a set of instructions.

```
while (condition):
    (instructions)
```

It executes instructions over and over until the condition is false, and when it is, it moves on to the next part of the code. For example:

```
import random
X=0
counter=0
while X==0:
    X=random.randrange(2)
    counter=counter+1
print counter
```

would simulate tossing a coin until it comes up "heads." (How come?) Make sure the condition becomes false eventually, or else the it will run forever!