

Problem Wk.10.1.1: Probability distributions: DDist

We will implement Python classes to represent and manipulate probability distributions over discrete sets. We will consider the [class DDist](#) that stores distributions in a [dictionary](#). You should review dictionaries (try, for example, the two Python Tutorial problems in Part 7).

Here is the basic code for the `DDist` class:

```
class DDist:
    def __init__(self, dictionary):
        self.d = dictionary

    def prob(self, elt):
        if self.d.has_key(elt):
            return self.d[elt]
        else:
            return 0

    def support(self):
        return [k for k in self.d.keys() if self.prob(k) > 0]
```

It is initialized with a dictionary whose keys are the values of the sample set of the distribution and whose values are probabilities. The values must always sum to 1.

The `prob` method takes an element and returns the probability assigned to it by the distribution. If the element is not explicitly represented in the distribution, it simply returns 0. This allows us to represent sparse distributions, in which only a few elements have non-zero probability, compactly.

The `support` method returns a list of all elements that have non-zero probability in this distribution.

Consider creating a distribution representing the probability of getting a particular grade in some hypothetical course:

```
>>> gradeDist = DDist({'a': 0.3, 'b': 0.3, 'c': 0.3, 'd': 0.07, 'f': 0.03})
DDist(a: 0.3, b: 0.3, c: 0.3, d: 0.07, f: 0.03)
```

Create a `DDist` that assigns probability 0.6 to the event 'hi', 0.1 to the event 'med', and 0.3 to the event 'lo'.

Enter an expression that sets the variable `foo` to the desired `DDist` instance. You do not need to say `dist.DDist`, it's enough to say `DDist`.

```
foo = None
```

MIT OpenCourseWare
<http://ocw.mit.edu>

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.