Sets, Multisets, Multimaps

- set: unordered collection of elements, no duplicates, typically supports efficient membership test
- multiset: set-like container that allows duplicates
- multimap: same key can be mapped to multiple values

↓

```
class Multimap:
  _MapType = dict
  def __init__(self):
    self._map = self._MapType()
    self._n = 0
  def __iter__(self):
    for k, secondary in self._map.items():
      for v in secondary:
        yield (k, v)
  def add(self, k, v):
    container = self._map.setdefault(k, [])
    container.append(v)
    self._n += 1
  def pop(self, k)
    secondary = self._map[k]
    v = secondary.pop()
    if len(secondary) == 0:
      del self._map[k]
    self._n -= 1
    return (k, v)
```

```
def find(self, k)
  secondary = self._map[k]
  return (k, secondary[0])
```

```
def find_all(self, k)
  secondary = self._map[k]
  [illegible]
  for v in secondary
    yield (k, v)
```

- designed so any map implementation can be used
- uses standard dict as map with list of values as composite value in dictionary