

CSC108H Lecture 21

Dan Zingaro

October 31, 2012

Example 1: Sum of List Values

```
def combine(d1, d2):  
    '''(dict of {object:list of int},  
    dict of {object:list of int}) ->  
    dict of {object:int}  
    Return the dictionary where each key is a key  
    that is in both d1 and d2.  
    The value associated with each key in the new  
    dictionary is the sum of all the integers associated  
    with that key in d1 and d2.  
  
    >>> combine({1:[2], 4:[5, 6]}, {4:[8]}  
    {4:19}  
    '''
```

Example 2: Folding Dictionaries

```
def fold(d1, d2):  
    '''(dict, dict) -> dict  
    Return a new dictionary that contains all (b, c)  
    such that (a, b) is in d1 and (a, c) is in d2.  
  
    >>> fold({1:4, 9:10}, {1:5})  
    {4:5}  
    '''
```

Is the folded dictionary **guaranteed** to be unique?

ConcepTest

the fish swim in the river

If we use a context of length 2, what is the context for swim?

- ▶ A. ('the', 'fish')
- ▶ B. ('in', 'the')
- ▶ C. ('fish', 'in')
- ▶ D. (oops, duplicate) ('the', 'fish')

ConceptTest

the birds fly in the sky

If we use a context of length 3, what is the context for in?

- ▶ A. ('birds', 'fly', 'in')
- ▶ B. ('birds', 'fly')
- ▶ C. ('the', 'birds', 'fly')
- ▶ D. ('the', 'birds', 'fly', 'in')

ConcepTest

```
t = ('cat', 'dog', 'mouse')  
t = t[1:] + ('wolf',)  
print(t)
```

What is the output of this code?

- ▶ A. ('dog', 'mouse', 'wolf')
- ▶ B. ('wolf', 'dog', 'mouse')
- ▶ C. ('dog', 'wolf')
- ▶ D. Error

Enhanced Version 2

Compared to version 1, version 2:

- ▶ Uses contexts of a specified length (not always length-one)
- ▶ Starts at a random context
- ▶ Chooses a random context instead of crashing when the context is not found in the dict