

## Lecture 24: Memoization

Given a list of integers L and an integer total, efficiently return a list of all pairs of numbers from L that total to total.

```
>>> pairs(list(range(10)), 7)
[(0, 7), (1, 6), (2, 5), (3, 4),
 (4, 3), (5, 2), (6, 1), (7, 0)]
```

A common practice in Information Retrieval is creating an inverse index. For example, suppose you had an index of words to their respective pages in the text in the form of a dictionary where the keys are words and the values are lists integers. Your goal is to create a dictionary where the keys are pages (integers), and the values are lists words.

```
>>> d = {str(chr(i)): [i,i+1,2*i] for i in range(68,72)}
```

```
>>> inverse(d)
```

```
{136: ['D'], 68: ['D'], 69: ['D', 'E'], 70: ['E', 'F'], 71: ['G', 'F'], 72: ['G
```

```
1 def memoize(obj):  
2     obj.data = {}  
3  
4     def memoizer(*args, **kwargs):  
5         key = str(args) + str(kwargs)  
6         if key not in obj.data:  
7             obj.data[key] = obj(*args, **kwargs)  
8         return obj.data[key]  
9  
10    return memoizer
```