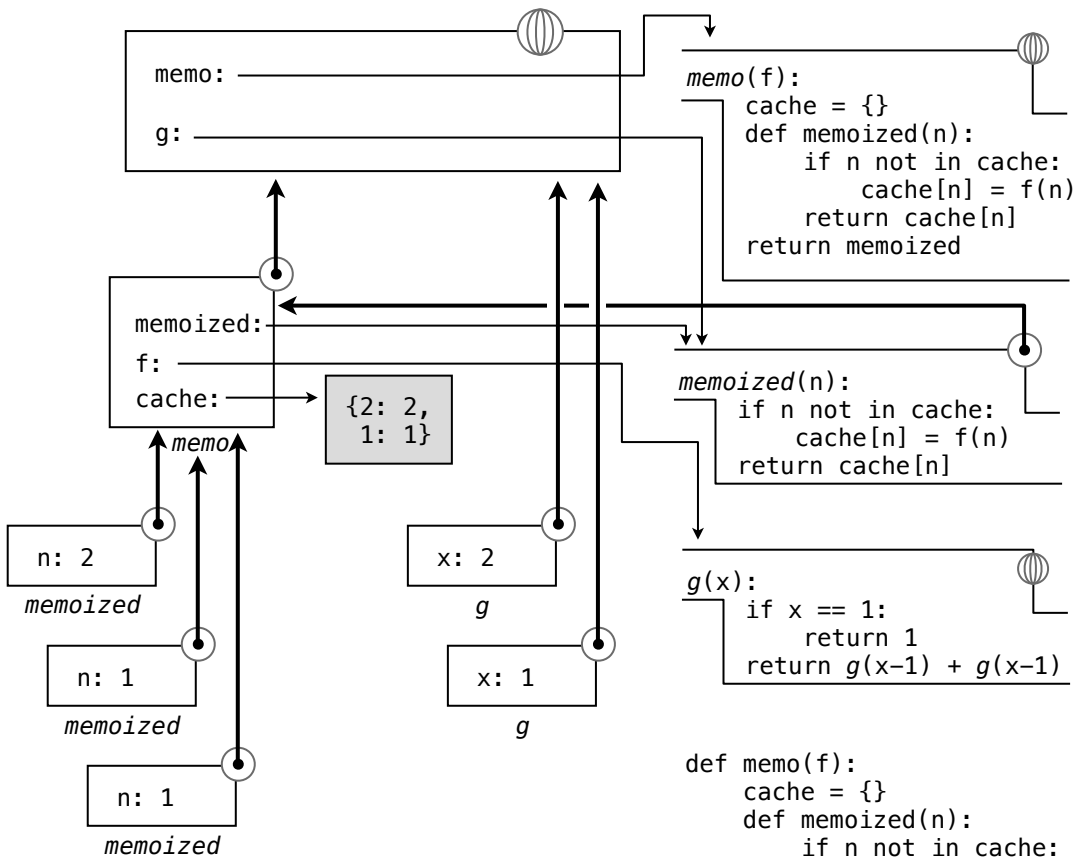


Environment Diagram for Memoize



```
memo(f):
  cache = {}
  def memoized(n):
    if n not in cache:
      cache[n] = f(n)
    return cache[n]
  return memoized
```

```
memoized(n):
  if n not in cache:
    cache[n] = f(n)
  return cache[n]
```

```
g(x):
  if x == 1:
    return 1
  return g(x-1) + g(x-1)
```

```
def memo(f):
  cache = {}
  def memoized(n):
    if n not in cache:
      cache[n] = f(n)
    return cache[n]
  return memoized
```

```
@memo
def g(x):
  if x == 1:
    return 1
  return g(x-1) + g(x-1)
```

g(2)

Notes:

- Memoized is called three times
- g is only called twice, once for each unique argument
- The gray box represents a dictionary; we don't have a "standard" way of depicting dictionaries in environment diagrams in this course.