BY: Zhu_Ming

<CS1010S>

Tutorial 9

Exception Handling, Memoization & Dynamic Programming

Lecture Recap

1. Dynamic Programming
2. Memoization
3. Exception Handling

Lecture_Recap.tut09

1. Dynamic Programming

- Bottom-Up approach
- Compute all value from bottom up
- The last value is the result

2. Memoization

- Top-Down approach
- Access previously calculated value
- If not calculated, compute and store in dictionary

3. Exception Handling

```
try:
   stuff
except ValueError: # catching an error
   print("I have ValueError!")
except Error as err: # storing error as err
   print("I have this error: " + str(err))
else: # if no error
   print("I have no error!")
   raise NameError("Error thrown on purpose")
finally: # will always be run
   print("just do it")
```

```
class LowCAPError(Exception):
   def __init__(self, cap):
      self.cap = cap

   def __str__(self):
      return str(cap) + " is too low!"
```

Important!!

If dealing with multiple exceptions, catch in **decreasing order of specificity.**

Your can also create your own custom exception

BY: Zhu_Ming

Any Questions?

Write a function collatz_distance(n) that returns the number of steps needed to reach 1 for an integer n.

$$f(x) = \begin{cases} n/2, & \text{if } \mathrm{even}(n) \\ 3n+1, & \text{if } \mathrm{odd}(n) \end{cases}$$

```
def collatz_distance(n):
   if n == 1:
      return 0

   elif n % 2 == 0:
      return 1 + collatz_distance(n//2)

   else:
      return 1 + collatz_distance(3*n + 1)
```

```
def collatz_distance(n):
   step = 0:

   while n != 1
      step += 1
      n = (n//2) if n % 2 == 0 else 3*n + 1
   return step
```

Write a function max_collatz_distance(n) that returns the maximum Collatz distance of 1, 2, 3, …, n.

$$f(x) = \begin{cases} n/2, & \text{if even}(n) \\ 3n+1, & \text{if odd}(n) \end{cases}$$

```
def max_collatz_distance(n):
   max_ = 0
   for i in range(1, n+1):
      curr = collatz_distance(i)
      max_ = curr if curr > max_ else max_
   return max_
```

```
def max_collatz_distance(n):
    return max(map(lambda num: collatz_distance(num), range(1,n+1)))
```

Question_3.tut09

Give a memoized version of max_collatz_distance_memo(n) using memoize as provided in the lecture.

```
memoize_table = {}

def memoize(f, name):
    if name not in memoize_table:
        memoize_table[name] = {}
        table = memoize_table[name]
        def helper(*args):
            if args in table:
                return table[args]
            else:
                result = f(*args)
                table[args] = result
                return result
    return helper
```

```
def collatz_distance_memo(n):
    def helper(n): # is same as collatz_distance
        if n == 1:
            return 0
        elif n % 2 == 0:
            return 1 + collatz_distance_memo(n//2)
        else:
            return 1 + collatz_distance_memo(3*n + 1)
    return memoize(helper, “memo_collatz”)(n)

def max_collatz_distance(n):
    return max(map(collatz_distance_memo, range(1, n + 1)))
```

Question_4.tut09

Memoize it without using the function provided in the lecture.

```python
def collatz_distance_memo(n):
   table = {1 : 0}
   def collatz(n): # is same as collatz_distance
      if n in table:
         return table[n]
      elif n % 2 == 0:
         table[n] = 1 + collatz(n//2)
      else:
         table[n] = 1 + collatz(3*n + 1)
      return table[n]
    return max(map(collatz, range(1, n + 1)))
```

Question_5.tut09

The following function accesses a URL on the internet and retrieves its contents:

```
from urllib.request import urlopen
from urllib.parse import urlsplit
from urllib.error import *

def httpget(url):
   parsed = urlsplit(url)
   if not parsed.scheme: # protocol insertion
      url = ‘http://’ + url
   elif parsed.scheme != ‘http’:
      raise ValueError(“Unknown protocol”)
   return urlopen(url).read()
```

Question_6.tut09

Your ability to access a URL on the internet is not guaranteed - it is only on a “best effort” basis. An example of a URL is http://www.nus.edu.sg/. Describe qualitatively some of the things that could go wrong.

```
URL spell wrong
Internet connection lousy
IMDA ban
Ethernet cable spoil
Satellite spoil
Server spoil
Singtel caught on fire
You use Singtel
… and more lol
```

Question_7.tut09

Why is it a good idea to raise an error instead of simply returning a string ‘Not Found’ or an empty string to indicate that the URL is not accessible?

It is possible that ‘Not Found’ or an empty string is an actual expected response from a server.

Modify httpget to accommodate error handling.

```
def httpget(url):
   parsed = urlsplit(url)
   if not parsed.scheme: # protocol insertion
      url = ‘http://’ + url
   elif parsed.scheme != ‘http’:
      raise ValueError(“Unknown protocol”)
   try:
      return urlopen(url).read()
   except HTTPError as err:
      raise InternetFail(“HTTPError - “ + str(err))
   except URLError as err:
      raise ValueError(str(err))
   except Exception as err:
      raise err
```

Important!!

Take note of the Exception Hierarchy

Question_9.tut09

Write a function download_URLs(URL_filenames) to download a set of files, where URL_filenames is a list of pairs [URL, filename]

```
def download_URLs(URL_filenames):
   for url, filename in URL_filenames:
      with open(filename, ‘wb’) as myFile:
      try:
         myFile.write(httpget(url))
      except (InternetFail, ValueError) as err:
         print(“could not get “+url+” :” + str(err))
      except Exception as err:
         raise err
```

Thank You!!

The End

See you next lesson

Tips.tut08

Q1: Algorithm question. revise for/while loops, if else blocks, recursion

Q2: Data processing. revise lists and HOF, will save you a lot of time. (mission 7)

Q3: OOP. revise everything related to OOP. (mission 12-14)

Don’t panic! Start from the one that you find it easy to do.
If stuck then move on to another question.
If have no idea, don’t leave it empty, write something make sense, let the grader to help you.

Good luck!!! :)

BY: Zhu_Ming

Any Questions?