

Python 递归出现的内存溢出

一、什么是内存溢出

内存溢出简单点说就是内存不够用了，当可使用的内存大于程序需要使用的内存时，就会发生内存溢出的情况。

二、递归如何导致内存溢出

函数调用是通过栈这种数据结构实现的，每分配一个栈就需要消耗一部分内存，在递归调用中，每当进入一个函数调用，栈就会加一层栈帧，每当函数返回，栈就会减一层栈帧。由于栈的大小不是无限的，所以，递归调用的次数过多，会导致栈溢出。如下例子：

1. 传递参数是 100 时，递归结果为 5050，可以正常算出结果

```
def sum2(n):
    if n > 0:
        return n + sum2(n - 1)
    else:
        return 0

print("递归结果-->", sum2(100))

sum1()
one (1) ×
"D:\Program Files\Python37\python.exe"
递归结果--> 5050
```

2. 当传递参数是 1000 时，递归系统报错，超过最大递归深度

```
def sum2(n):
    if n > 0:
        return n + sum2(n - 1)
    else:
        return 0

print("递归结果-->",sum2(1000))

sum1()
one (1) ✘
  File "D:/my/code/one.py", line 14, in sum2
    if n > 0:
RecursionError: maximum recursion depth exceeded in comparison
```