深度优先和广度优先

Python 实现深度优先

def depth\_tree(tree\_node):

if tree\_node.\_left is not None:

return depth\_tree(tree\_node\_left)

if tree\_node.\_right is not None:

return depth\_tree(tree\_node.\_right)

广度优先遍历

def level\_queue(root):

if root is None:

return

my\_queue = []

node = root

my\_queue.append(node)

while my\_queue:

node = my\_queue.pop(0)

print (node.elem)

if node.lchild is not None:

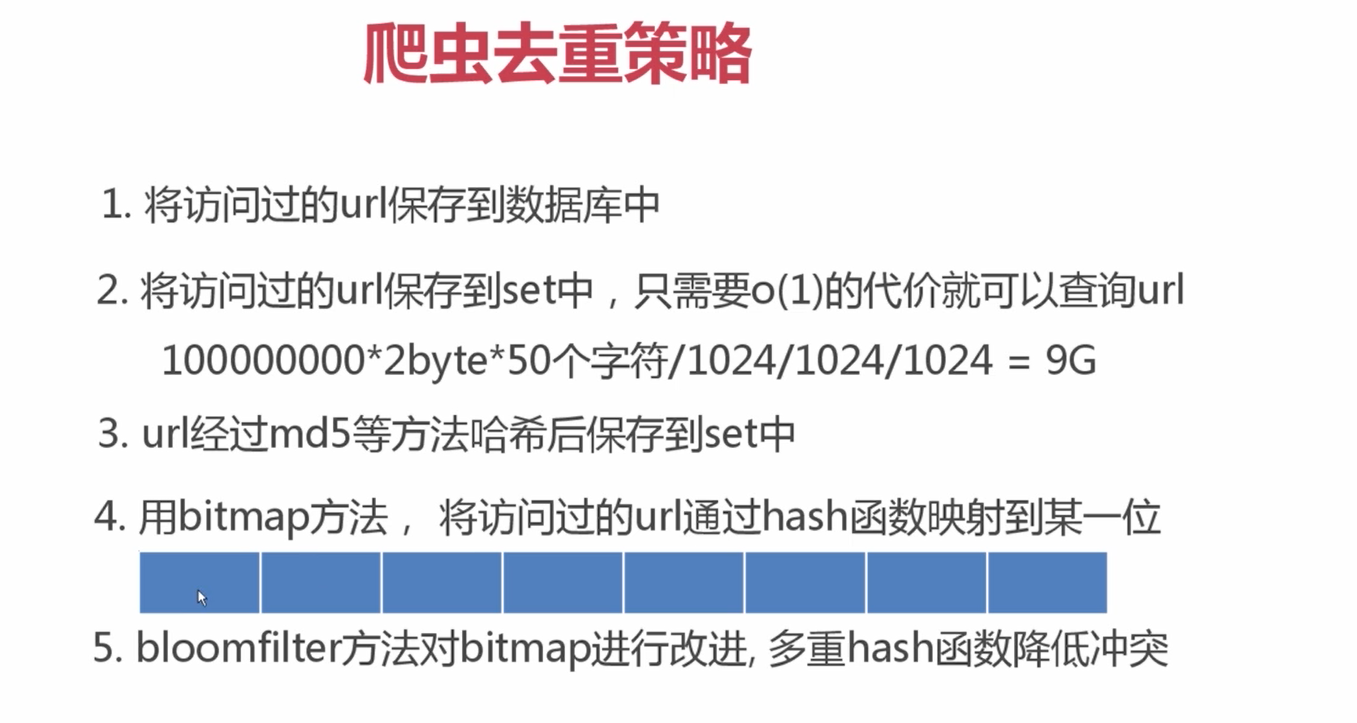
my\_queue.append(node.lchild)

if node.rchild is not None:

my\_queue.append(node.rchild)

url 去重

url去重的策略



数据量大的时候用bloomfilter去重是效果非常明显的，scrapy一般用的第三种

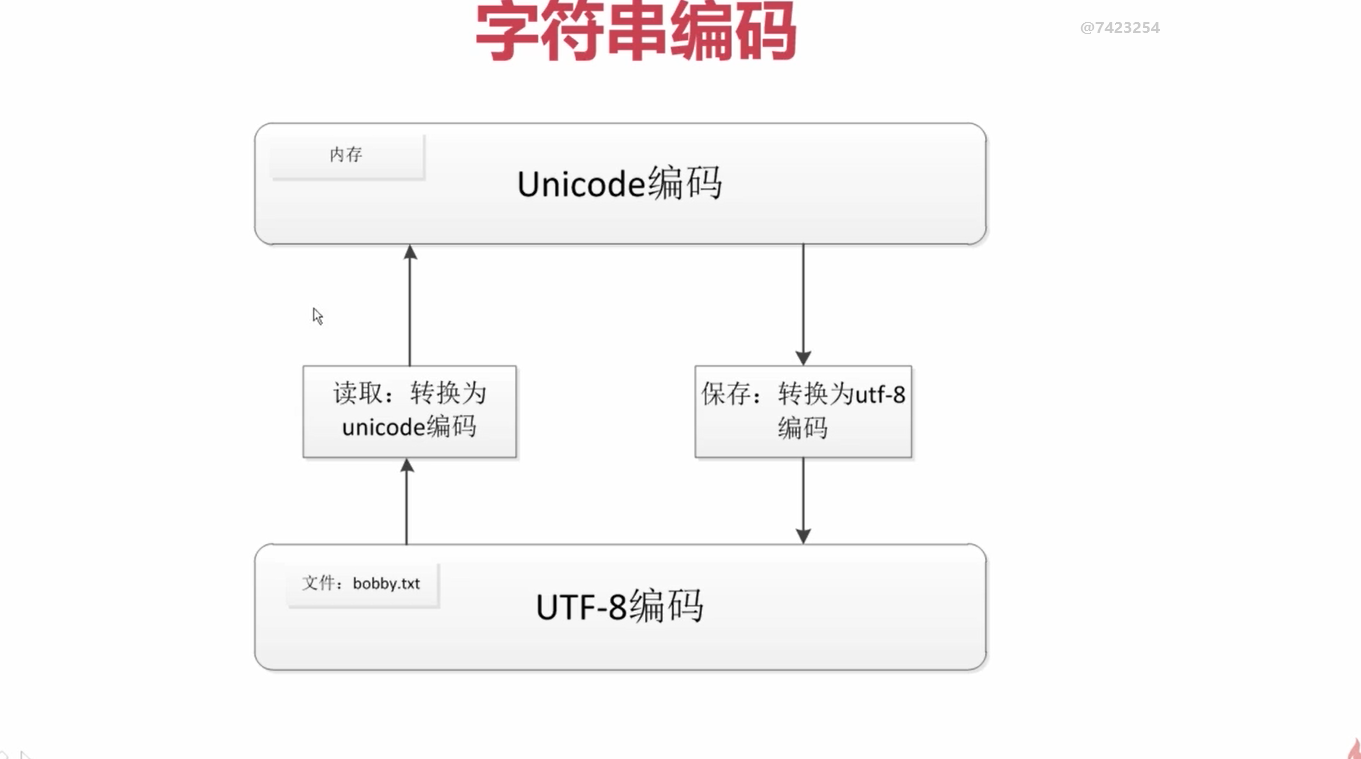
8bit = 1字节

Unicode编码

解决编码范围

问题是若是英文则血亏，多占1倍内存

Utf-8此时出现了作用



Python3解决编码问题：

S=”我用Python“

S.encode(“utf-8”)#意思就是i先转化为unicode再变