Python Suffix Stripping Stemmer Hackerrank Solution

```python
1    #!/bin/python3
2
3    import math
4    import os
5    import random
6    import re
7    import sys
8
9
10   n = int(input().strip())
11
12   if __name__ == '__main__':
13       if n % 2 != 0:
14           print("Weird")
15       else:
16           if n in range(2, 5):
17               print("Not Weird")
18           elif n >= 6 and n <= 20:
19               print("Weird")
20           elif n > 21:
21               print("Not Weird")
22
23
```

Python suffix stripping stemmer HackerRank solution is a fascinating topic that delves into the world of natural language processing (NLP) and text analysis. Stemming is a crucial technique in NLP that involves reducing words to their base or root form. This is particularly useful in various applications, including information retrieval, text mining, and search engines. In this article, we will explore the concept of suffix stripping, how to implement a stemmer in Python, and provide an example solution commonly found in HackerRank challenges.

Understanding Stemming and Its Importance

Stemming refers to the process of removing suffixes from words to retrieve their base or root form. For example, "running," "runner," and "ran" can all be reduced to the root word "run." This technique is vital for several reasons:

- **Improved Search Results:** By reducing words to their stems, search algorithms can return more relevant results based on the root word rather than the specific forms of

the word.

- **Data Normalization:** Stemming helps in normalizing text data, making it easier to analyze and process.

- **Reducing Dimensionality:** In text classification and clustering, stemming helps reduce the number of unique words, simplifying the model complexity.

Types of Stemming Algorithms

There are various stemming algorithms in NLP, but the two most commonly used are:

1. Porter Stemmer

Developed by Martin Porter in 1980, the Porter Stemmer is one of the most popular algorithms for stemming. It applies a series of rules to strip suffixes from words. The process is relatively simple but effective for many English words.

2. Snowball Stemmer

An improvement on the Porter Stemmer, the Snowball Stemmer provides a more sophisticated approach to stemming. It includes support for multiple languages and offers a more extensive set of rules for suffix stripping.

Implementing a Suffix Stripping Stemmer in Python

To create a suffix stripping stemmer in Python, we will focus on the Porter Stemmer as it is widely used for educational purposes and is straightforward to implement. Below are the steps to create a simple stemmer:

Step 1: Define the Suffixes

The first step is to define the suffixes that we want to strip from the words. Here's a basic list of common English suffixes:

```python
suffixes = [
```

```
'ing', 'ed', 'ly', 'es', 's',
'ment', 'ness', 'ful', 'able',
'ible', 'tion', 'ation', 'al',
'ic', 'er', 'or', 'ism', 'ist',
'ity', 'ty', 'y', 'e'
]
```

Step 2: Create the Stemmer Function

Next, we will create a function that takes a word as input and removes the defined suffixes:

```python
def stem(word):
for suffix in suffixes:
if word.endswith(suffix):
return word[:-len(suffix)]
return word
```

In this function, we iterate through the list of suffixes. If the word ends with a specific suffix, we remove it and return the stemmed word.

Step 3: Testing the Stemmer

We can now test our stemmer function with a list of example words:

```python
words = ['running', 'happiness', 'played', 'quickly', 'friendship']
stemmed_words = [stem(word) for word in words]
print(stemmed_words)
```

This code will output the stemmed version of the input words. However, note that this simple implementation may not handle all cases perfectly, as stemming can be context-dependent.

HackerRank Challenge: Suffix Stripping Stemmer

On platforms like HackerRank, challenges often require participants to implement a stemmer efficiently. Here's how you might approach a HackerRank-style problem involving suffix stripping.

Problem Statement

You are given a list of words and your task is to return a list of their stemmed forms using a suffix stripping algorithm.

Sample Input

```python
input_words = ['studies', 'studying', 'study', 'running', 'runner']
```

Sample Output

```python
['studi', 'studi', 'studi', 'run', 'run']
```

Solution Implementation

Here's a complete solution that incorporates the stemmer function into a HackerRank-style problem:

```python
def stem(word):
suffixes = [
'ing', 'ed', 'ly', 'es', 's',
'ment', 'ness', 'ful', 'able',
'ible', 'tion', 'ation', 'al',
'ic', 'er', 'or', 'ism', 'ist',
'ity', 'ty', 'y', 'e'
]

for suffix in suffixes:
if word.endswith(suffix):
return word[:-len(suffix)]
return word

def process_words(words):
return [stem(word) for word in words]

Example usage
input_words = ['studies', 'studying', 'study', 'running', 'runner']
output = process_words(input_words)
print(output) Output: ['studi', 'studi', 'studi', 'run', 'run']
```

```

# Conclusion

In this article, we have explored the concept of suffix stripping and its application in stemming words using Python. The Python suffix stripping stemmer HackerRank solution demonstrates a practical approach to solving stemming problems, providing a foundation for further exploration in natural language processing. By understanding how to remove suffixes effectively, you can enhance your text analysis capabilities and improve the performance of various applications in data science and machine learning.

As you continue to practice and challenge yourself with coding exercises, remember that variations of this basic stemmer can be expanded with more sophisticated techniques, such as using regular expressions or integrating machine learning models for better accuracy. Happy coding!

# Frequently Asked Questions

## What is a suffix stripping stemmer in Python?

A suffix stripping stemmer in Python is a tool used in natural language processing that reduces words to their base or root form by removing suffixes, thus allowing for better information retrieval and text analysis.

## How does the suffix stripping stemmer work in the HackerRank challenge?

In the HackerRank challenge, the suffix stripping stemmer typically involves writing a function that takes a word as input and removes specific suffixes based on predefined rules to return the stemmed version of the word.

## What are common suffixes that a suffix stripping stemmer might remove?

Common suffixes include 'ing', 'ed', 'ly', 's', 'es', and 'tion'. The stemmer uses these suffixes to identify and strip them from the end of words.

## Can you provide a simple example of a suffix stripping function in Python?

Certainly! A simple example could be: `def stem(word): return word[:-3] if word.endswith('ing') else word`. This removes 'ing' from the end of the word if it exists.

# What is the significance of stemming in text processing?

Stemming is significant in text processing as it helps in reducing words to their root form, which can improve the accuracy of search queries and the efficiency of text analysis by grouping similar words together.

# How can I test my suffix stripping stemmer on HackerRank?

You can test your suffix stripping stemmer on HackerRank by submitting your function as a solution and then running the provided test cases to ensure it behaves as expected with various inputs.

# What are the limitations of suffix stripping stemmers?

Limitations of suffix stripping stemmers include potential over-stemming, where a stemmer may reduce different words to the same root incorrectly, and under-stemming, where it fails to reduce words that should be stemmed.

# How does the suffix stripping approach differ from lemmatization?

Suffix stripping is a more aggressive approach that removes suffixes to find the stem, while lemmatization considers the context and converts a word to its meaningful base form, often using a dictionary.

# What Python libraries can help with stemming?

Python libraries such as NLTK (Natural Language Toolkit) and Snowball stemmer provide built-in functions for stemming, including suffix stripping algorithms.

# What is the expected output of a correctly implemented suffix stripping stemmer?

The expected output of a correctly implemented suffix stripping stemmer is the base form of input words with suffixes removed, such as converting 'running' to 'run' or 'happily' to 'happi'.

Find other PDF article:

# Python Suffix Stripping Stemmer Hackerrank Solution

*What does colon equal (:=) in Python mean? - Stack Overflow*
Mar 21, 2023 · In Python this is simply =. To translate this pseudocode into Python you would need to know the data structures being referenced, and a bit more of the algorithm implementation. Some notes about psuedocode: := is the assignment operator or = in Python = is the equality operator or == in Python There are certain styles, and your mileage may vary:

## What does asterisk * mean in Python? - Stack Overflow
What does asterisk * mean in Python? [duplicate] Asked 16 years, 7 months ago Modified 1 year, 6 months ago Viewed 319k times

## What does the "at" (@) symbol do in Python? - Stack Overflow
Jun 17, 2011 · 96 What does the "at" (@) symbol do in Python? @ symbol is a syntactic sugar python provides to utilize decorator, to paraphrase the question, It's exactly about what does decorator do in Python? Put it simple decorator allow you to modify a given function's definition without touch its innermost (it's closure).

Is there a "not equal" operator in Python? - Stack Overflow
Jun 16, 2012 · 1 You can use the != operator to check for inequality. Moreover in Python 2 there was <> operator which used to do the same thing, but it has been deprecated in Python 3.

## Using or in if statement (Python) - Stack Overflow
Using or in if statement (Python) [duplicate] Asked 7 years, 6 months ago Modified 8 months ago Viewed 149k times

*python - What is the purpose of the -m switch? - Stack Overflow*
Python 2.4 adds the command line switch -m to allow modules to be located using the Python module namespace for execution as scripts. The motivating examples were standard library modules such as pdb and profile, and the Python 2.4 implementation is …

*What is Python's equivalent of && (logical-and) in an if-statement?*
Mar 21, 2010 · There is no bitwise negation in Python (just the bitwise inverse operator ~ - but that is not equivalent to not). See also 6.6. Unary arithmetic and bitwise/binary operations and 6.7. Binary arithmetic operations. The logical operators (like in many other languages) have the advantage that these are short-circuited.

syntax - What do >> and <
Apr 3, 2014 · 15 The other case involving print >>obj, "Hello World" is the "print chevron" syntax for the print statement in Python 2 (removed in Python 3, replaced by the file argument of the print() function). Instead of writing to standard output, the output is passed to the obj.write() method. A typical example would be file objects having a write() method.

## **python - Is there a difference between "==" and "is"? - Stack …**
Since is for comparing objects and since in Python 3+ every variable such as string interpret as an object, let's see what happened in above paragraphs. In python there is id function that shows a unique constant of an object during its lifetime. This id is using in back-end of Python interpreter to compare two objects using is keyword.

python - What does ** (double star/asterisk) and * (star/asterisk) …
Aug 31, 2008 · A Python dict, semantically used for keyword argument passing, is arbitrarily ordered. However, in Python 3.6+, keyword arguments are guaranteed to remember insertion order.

**What does colon equal (:=) in Python mean? - Stack Overflow**
Mar 21, 2023 · In Python this is simply =. To translate this pseudocode into Python you would need to know the data structures being referenced, and a bit more of the algorithm implementation. Some notes about psuedocode: := is the assignment operator or = in Python = is the equality operator or == in Python There are certain styles, and your mileage may vary:

**What does asterisk * mean in Python? - Stack Overflow**
What does asterisk * mean in Python? [duplicate] Asked 16 years, 7 months ago Modified 1 year, 6 months ago Viewed 319k times

What does the "at" (@) symbol do in Python? - Stack Overflow
Jun 17, 2011 · 96 What does the "at" (@) symbol do in Python? @ symbol is a syntactic sugar python provides to utilize decorator, to paraphrase the question, It's exactly about what does decorator do in Python? Put it simple decorator allow you to modify a given function's definition without touch its innermost (it's closure).

*Is there a "not equal" operator in Python? - Stack Overflow*
Jun 16, 2012 · 1 You can use the != operator to check for inequality. Moreover in Python 2 there was <> operator which used to do the same thing, but it has been deprecated in Python 3.

*Using or in if statement (Python) - Stack Overflow*
Using or in if statement (Python) [duplicate] Asked 7 years, 6 months ago Modified 8 months ago Viewed 149k times

**python - What is the purpose of the -m switch? - Stack Overflow**
Python 2.4 adds the command line switch -m to allow modules to be located using the Python module namespace for execution as scripts. The motivating examples were standard library modules such as pdb and profile, and the Python 2.4 implementation is ...

What is Python's equivalent of && (logical-and) in an if-statement?
Mar 21, 2010 · There is no bitwise negation in Python (just the bitwise inverse operator ~ - but that is not equivalent to not). See also 6.6. Unary arithmetic and bitwise/binary operations and 6.7. Binary arithmetic operations. The logical operators (like in many other languages) have the advantage that these are short-circuited.

**syntax - What do >> and <**
**Apr 3, 2014 · 15 The other case involving print >>obj, "Hello World" is the "print chevron" syntax for the print statement in Python 2 (removed in Python 3, replaced by the file argument of the print() function). Instead of writing to standard output, the output is passed to the obj.write() method. A typical example would be file objects having a write() method.**

**python - Is there a difference between "==" and "is"? - Stack ...**
**Since is for comparing objects and since in Python 3+ every variable such as string interpret as an object, let's see what happened in above paragraphs. In python there is id function that shows a unique constant of an object during its lifetime. This id is using in back-end of Python interpreter to compare two objects using is keyword.**

*python - What does ** (double star/asterisk) and * (star/asterisk) ...*
**Aug 31, 2008 · A Python dict, semantically used for keyword argument passing, is arbitrarily ordered. However, in Python 3.6+, keyword arguments are guaranteed to**

**remember insertion order.**

**Master the Python suffix stripping stemmer with our comprehensive HackerRank solution guide. Learn how to enhance your coding skills and tackle challenges effectively!**

[Back to Home](#)