Faster Pattern Avoidance

API Documentation

July 29, 2013

Contents

1 Script script-pattern_spyx

A Pattern module for Sage Mathematics

Contains fast implementations for methods Patt.subwords using dynamic programming and Patt.avoid by reducing redundant computations.

1.1 Classes

- **Pattern**: A Classical Pattern
  (Section 3, p. 4)
- **LengthPattern**: A Classical Pattern and a specific length of permutations
  (Section 2, p. 3)

1.2 Functions

flatten_(*lst*)

Return a flattened list

Slow implementation, O(n^2)

```
>>> flatten_(3, 8, 2)
[2,3,1]
```

2 Class script-pattern_spyx.LengthPattern

object
└ **script-pattern_spyx.LengthPattern**

A Classical Pattern and a specific length of permutations

2.1 Methods

__init__(*self*, *patt*, *perm_len*)

Initialise LengthPattern with a classical pattern patt (list) and integer length perm_len (int)

```
>>> LengthPattern([1,2,3,4], 10)
LengthPattern [1, 2, 3, 4] with length 10
```

Overrides: object.__init__

subwords(*self*, *perm*)

Generator for subwords of perm that contain Patt

Fast implementation, might allocate a lot of memory

```
>>> [perm for perm in LengthPattern([1,2,3], 4).subwords([1,2,3,4])]
[[2, 3, 4], [1, 2, 3], [1, 2, 4], [1, 3, 4]]
```

__str__(*self*)

str(x)

Overrides: object.__str__ extit(inherited documentation)

__repr__(*self*)

repr(x)

Overrides: object.__repr__ extit(inherited documentation)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __subclasshook__()

2.2 Properties

Name	Description
Inherited from object	
__class__	

3 Class script-pattern_spyx.Pattern

object
└ **script-pattern_spyx.Pattern**

A Classical Pattern

3.1 Methods

__init__(*self*, *pattern*)

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

clear_cache(*self*)

Clear memory allocated by self

```
>>> p.clear_cache()
Cache cleared
```

get_patt(*self*, *perm*)

Patt object with corresponding length for perm

```
>>> Pattern([1,2,3]).get_patt([1,2,3,4])
LengthPattern [1, 2, 3] with length 4
```

subwords(*self*, *perm*)

Generator for subwords of perm containing Pattern

```
>>> [perm for perm in Pattern([1,2,3]).subwords([1,5,2,4,3])]
[[1, 2, 4], [1, 2, 3]]
```

subwords_list(*self*, *perm*)

List of subwords of perm containing Pattern

```
>>> Pattern([1,2,3]).subwords_list([1,5,2,4,3])
[[1, 2, 4], [1, 2, 3]]
```

subwords_print(*self*, *perm*)

Print out subwords of perm containing Pattern

avoided_by(*self*, *perm*)

True if perm avoids pattern, false otherwise

```
>>> Pattern([1,2,3]).avoided_by([4,3,1,2,5])
False
>>> Pattern([1,2,3]).avoided_by([5,4,3,1,2])
True
```

avoiders(*self*, *n*)

Generator for permutations of length n that avoid Pattern

Fast implementations, it does not interate through all permuatation of length n by default. Instead, it begins with all permutations one larger than the given pattern and then expands only those permutations that avoid the pattern.

```
>>> [perm for perm in Pattern([1,2,3]).avoiders(4)]
[[1, 4, 3, 2],
[2, 1, 4, 3],
[2, 4, 1, 3],
[2, 4, 3, 1],
[3, 1, 4, 2],
[3, 2, 1, 4],
[3, 2, 4, 1],
[3, 4, 1, 2],
[3, 4, 2, 1],
[4, 1, 3, 2],
[4, 2, 1, 3],
[4, 2, 3, 1],
[4, 3, 1, 2],
[4, 3, 2, 1]]
```

avoiders_cardinality(*self*, *n*)

Count the number of permutations of length n that avoid Pattern

```
>>> Pattern([1,2,3]).avoiders_cardinality(10)
16796
```

avoiders_list(*self*, *n*)

List of permutations of length n that avoid Pattern

```
>>> Pattern([1,2,3]).avoiders_list(4)
[[1, 4, 3, 2],
[2, 1, 4, 3],
[2, 4, 1, 3],
[2, 4, 3, 1],
[3, 1, 4, 2],
[3, 2, 1, 4],
[3, 2, 4, 1],
[3, 4, 1, 2],
[3, 4, 2, 1],
[4, 1, 3, 2],
[4, 2, 1, 3],
[4, 2, 3, 1],
[4, 3, 1, 2],
[4, 3, 2, 1]]
```

expanded(*self*, *perm*)

Generator for exansions of perm

```
>>> [perm for perm in expand([1,2,3])]
[[4, 1, 2, 3], [1, 4, 2, 3], [1, 2, 4, 3], [1, 2, 3, 4]]
```

__**str**__(*self*)

str(x)

Overrides: object.__str__ extit(inherited documentation)

__**repr**__(*self*)

repr(x)

Overrides: object.__repr__ extit(inherited documentation)

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __setattr__(), __sizeof__(), __subclasshook__()

3.2 Properties

Name	Description
Inherited from object	
__class__	

Index