

回文数

生成回文数

方式一

```
In[*]:= f1[n_] := Select[Range[0, 10^n - 1], PalindromeQ]
           |选择   |范围   |回文判定

Table[f1[i], {i, 1, 3}] // Column
|表格   |列
Out[*]=
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99}
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101, 111, 121, 131, 141,
 151, 161, 171, 181, 191, 202, 212, 222, 232, 242, 252, 262, 272, 282, 292, 303, 313,
 323, 333, 343, 353, 363, 373, 383, 393, 404, 414, 424, 434, 444, 454, 464, 474, 484,
 494, 505, 515, 525, 535, 545, 555, 565, 575, 585, 595, 606, 616, 626, 636, 646, 656,
 666, 676, 686, 696, 707, 717, 727, 737, 747, 757, 767, 777, 787, 797, 808, 818, 828,
 838, 848, 858, 868, 878, 888, 898, 909, 919, 929, 939, 949, 959, 969, 979, 989, 999}
```

```
In[*]:= Length[123]
           |长度
Out[*]=
0
```

数字长度

```
In[*]:= DigitCount[12345]
           |数字个数
Out[*]=
{1, 1, 1, 1, 1, 0, 0, 0, 0, 0}
```

```
In[*]:= IntegerLength[12345]
           |整数长度
Out[*]=
5
```

三角矩阵

不使用Flatten和Sort，当然这并不是最快的

```
In[*]:= RepeatedTiming[SelectFirst[# == IntegerReverse@# &] @
           |计算表达式的平... |选择第一个 |颠倒整数
           Reverse@Apply[Union, #^T.# &@{Range[100, 999]}], 1]
           |反向排序 |应用 |并集 |范围
Out[*]=
{0.0478595, 906609}
```

```

In[ ]:= RepeatedTiming[
  |计算表达式的平均时间
  SelectFirst[PalindromeQ]@Reverse@Apply[Union, #^.# &@{Range[100, 999]}], 1]
  |选择第一个 |回文判定 |反向排序 |应用 |并集 |范围
Out[ ]:=
{0.0480533, 906 609}

In[ ]:= RepeatedTiming[SelectFirst[# == IntegerReverse@# &]@
  |计算表达式的平... |选择第一个 |颠倒整数
  Reverse@Apply[Union, m^m /. m -> {Range[100, 999]}], 1]
  |反向排序 |应用 |并集 |范围
Out[ ]:=
{0.0466563, 906 609}

In[ ]:= RepeatedTiming[
  |计算表达式的平均时间
  SelectFirst[PalindromeQ]@Reverse@Apply[Union, m^m /. m -> {Range[100, 999]}], 1]
  |选择第一个 |回文判定 |反向排序 |应用 |并集 |范围
Out[ ]:=
{0.0466786, 906 609}

```

结论：
替换比自定义函数快一点点