

Project Euler 36. Double-base Palindromes

hiragn

2024 年 12 月 21 日

1. 問題の概要

$585 = 1001001001_{(2)}$ 10 進法でも 2 進法でも回文数である。

100 万未満で 10 進法でも 2 進法でも回文数になる数の総和を求めよ。

注) リーディングゼロは考えない。

<https://projecteuler.net/problem=36>

2. PalindromeQ を使う

回文数を判定する関数 `PalindromeQ` を利用します。

`PalindromeQ` の引数はリスト, 10 進整数, 文字列のどれかです。2 進法の回文判定はそのままではできないので, `IntegerDigits[#, 2]` で 2 進法表示したときの各位の数字のリストを作って `PalindromeQ` に渡します。

条件をみたす数を 2 進法であらわしたとき, 首位の数字は 1 なので末位も 1 です。100 万未満の奇数について調べました。

```
1 In[] := Clear["Global`*"];
2 RepeatedTiming[
3   cond[n_] := PalindromeQ@n && PalindromeQ@IntegerDigits[n, 2];
4   ans = Total@Parallelize@Select[Range[1, 10^6, 2], cond]]
5
6 Out[] = {0.630105, 872187}
```

3. 左半分を決める

回文数の対称性を利用します。

- 回文数の右半分の桁は左半分の桁を反転したもの
- 100 万の 2 進法表示は 20 桁

$2^{10} \doteq 1000$ なので 2 進法であらわしたときの左半分の桁は約 1,000 通りしかありません。これらについて調べます。

1. 各 n を 2 進法表示したときの各位の数字のリスト L を作る
2. L を反転したリスト R を作る
3. $L + R$, $L + \{0\} + R$, $L + \{1\} + R$ のようにリストを連結して 10 進法に直す
4. 回文数の判定

1 だけはこの方法で作れないので特別扱いが必要です。

この方法で計算すると PalindromeQ 頼みの解法の 60 倍くらい速いです。

```

1 In[] := Clear["Global`*"];
2 RepeatedTiming[
3   nmax = 10^6;
4   cond[x_] := x <= nmax && PalindromeQ@x;
5   calc[n_] := Module[{lft, right, r, nums},
6     lft = IntegerDigits[n, 2];
7     right = Reverse@lft;
8     nums = FromDigits[#, 2] & /@
9       {Join[lft, right], Join[lft, {0}, right], Join[lft, {1}, right]};
10    Total@Select[nums, cond]];
11 ans = 1 + Total[calc /@ Range@Floor@Sqrt@nmax]]
12
13 Out[] = {0.012219, 872187}

```
