

# 連続する 2 桁がすべて 7 で割りきれ数

hiragn

2024 年 12 月 12 日

「算数にチャレンジ!!」第 1298 回の問題<sup>\*1</sup>解いた。

2149 は連続する 2 桁 (21, 14, 49) がすべて 7 の倍数になっている。  
このような整数の最大値を求めよ。各位の数はすべて異なり、また、どの位にも 0 は使わないものとする。

<http://www.sansu.org/used-html/index1298.html>

## 1. 手計算で解くなら

普通に手計算で解くなら、まずは 2 桁の 7 の倍数のうち十の位と一の位が異なっていて、どちらも 0 でないものを書き出すところからはじめるはず。

14, 21, 28, 35, 42, 49, 56, 63, 84, 91, 98

このリストをもとに  $14 \rightarrow 42 \rightarrow 28 \rightarrow \dots$  のように各数の末尾と先頭をつないでいくと条件をみたす数を全部作ることができて、最大値は 98421 だとわかる。

「矢印のかわりに有向辺で結べば、グラフの問題になりそう」と思ったのでやってみた。  
また、「グラフを使うのはさすがに大げさかな」とも思ったので、dfs でも解いた。

---

<sup>\*1</sup> この問題文は原題を適当に書き換えたもの。

## 2. グラフを使う

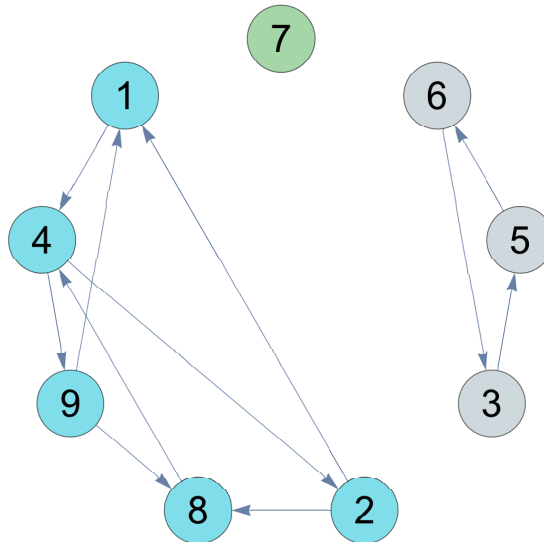
まず,  $10i + j$  が 7 で割りきれられるような  $i$  と  $j$  を有向辺で結んでグラフを作る。

これは RelationGraph 一発でできて, 次のようになる。

---

```
1 cond[x_, y_] := Divisible[10 x + y, 7] && x != y;
2 g = RelationGraph[cond, Range@9]
```

---



7 が孤立していること, ループを含むことがわかる。

ループを含むので「辺のコストを  $-1$  倍して最短距離の問題に言い換える」みたいなことはできない。「2 つの頂点を結ぶ経路を全部求めて, 対応する数の最大値を求める」で解く。

出発点とゴールの組み合わせは全部で  ${}_9C_2 = 45$  通り。これを全部調べても大した時間はかからないが, 弱連結 (無向グラフとみなしたときに連結) な成分に分割すると計算量を減らせる。

弱連結成分分解は WeaklyConnectedComponents で簡単にできて, 頂点のリストが返ってくる。

---

```
1 In[] := sg = Select[WeaklyConnectedComponents@g, Length@# > 1 &]
2
3 Out[] = {{1, 4, 2, 9, 8}, {3, 5, 6}}
```

---

経路の探索には FindPath を使う。引数に Infinity と All を指定して, 2 点間の経路を長さ と本数の制限なしに全部求めると計算時間は約 0.0009 秒。

---

```

1 In[] := Clear["Global`*"];
2 AbsoluteTiming[
3   cond[x_, y_] := Divisible[10 x + y, 7] && x != y;
4   g = RelationGraph[cond, Range@9, VertexLabels -> Automatic];
5   sg = Select[WeaklyConnectedComponents@g, Length@# > 1 &];
6   calc[lst_] :=
7     Max[FromDigits /@ FindPath[g, #1, #2, Infinity, All] & @@@
8       Tuples[lst, 2]];
9   ans = Max[calc /@ sg]]
10
11 Out[] = {0.000884, 98421}

```

---

### 3. dfs する

「普通に手計算で解くなら～」で書いた方法を自動化すると dfs になる。

「 $n$  の桁数字に含まれない数  $k$  に対して  $\text{dfs}(10n + k)$ 」を繰り返しつつ  $n$  の最大値を更新していく。 $k$  の選択は  $n$  の桁数字の集合と  $\{1, 2, \dots, 9\}$  の補集合 (Complement) をとることによって簡単にできた。

計算時間は約 0.0007 秒で、グラフを使うよりこちらの方が速かった。

---

```

1 In[] := Clear["Global`*"];
2 AbsoluteTiming[
3   ans = 0;
4   dfs[n_] := Module[{digits},
5     ans = Max[ans, n];
6     digits = Select[Complement[Range@9, IntegerDigits@n],
7       Divisible[10*Mod[n, 10] + #, 7] &];
8     dfs[10 n + #] & /@ digits];
9   dfs /@ Range@9;
10  ans]
11
12 Out[] = {0.000717, 98421}

```

---