# Project Euler 60. Prime Pair Sets

hiragn

2024 年 12 月 26 日

## 1. 問題の概要

素数 3, 7, 109, 673 には面白い性質がある。任意の 2 つの素数を任意の順でつなげると素数になる。たとえば 7 と 109 からできる 7109 と 1097 は両方とも素数である。

これら 4 つの素数 3, 7, 109, 673 の和は 792 である。これはこのような性質をもつ 4 つの素数の集合の和の中で最小である。

任意の 2 つの素数をつなげたときに別の素数ができる 5 つの素数の集合の和の最小値を求めよ。

`https://projecteuler.net/problem=60`

## 2. 解を 1 つ求める

適当に上限を決めてグラフ理論を使ったら意外にすんなり解がみつかりました。

1. 連結した数が 2 つとも素数になる素数 $p$, $q$ を辺で結んでグラフを作る
2. 5 個の点からなるクリーク（フルメッシュな部分グラフ）を探す
3. その頂点に対応する素数の和を求める

計算量を減らすため，以下のような工夫をしています。

- 素数 2, 5 は使わない。これらが 1 の位に来たら素数でなくなるため
- 3 で割った余りが 1 の数と 2 の数は連結しない。連結すると 3 の倍数ができて，それは素数ではない

```
In[]:= Clear["Global‘*"];
RepeatedTiming[
 (* 素数のペアを作る *)
 nmax = 10000;
 asc = GroupBy[Prime@Range[4, PrimePi@nmax], Mod[#, 3] &];
 pairs = Flatten[Subsets[#, {2}] & /@
    {Join[asc@1, {3}], Join[asc@2, {3}]}, 1];

 (* 条件をみたす素数を辺で結んでグラフを作る *)
 dgt[i_] := dgt[i] = IntegerDigits@i;
 cond[{i_, j_}] := PrimeQ@FromDigits@Join[dgt@i, dgt@j];
 makeEdge[{i_, j_}] := If[AllTrue[{{i, j}, {j, i}}, cond],
   UndirectedEdge[i, j], Nothing];
 gr = Graph[makeEdge /@ pairs, VertexLabels -> Automatic];

 (* 5個の頂点からなるクリークを探す *)
 ans = Min[Total /@ FindClique[gr, {5}, All]]]

Out[]= {0.964127, 26033}
```

これで正解なのですが，上限の 10,000 は試行錯誤で見つけただけで，この範囲に解があることは明らかではありません。また，

$$26033\ (= 13 + 5197 + 5701 + 6733 + 8389)$$

が最小値であることも明らかではありません。5 個目の素数が大きくても和が小さいケースも考えられます。

# 3. 正当化する

## 3.1 検索範囲の上限

素数の上限については「頂点数 5 のクリークが生じるまで頂点を 1 個ずつ追加し続ける」で調べられます。1051 個目の素数 8389 まで調べれば十分でした。

```
In[]:= Clear["Global‘*"];
AbsoluteTiming[
 (* 2つの素数を辺で結ぶ条件 *)
 dgt[i_] := dgt[i] = IntegerDigits@i;
 cond1[{i_, j_}] := PrimeQ@FromDigits@Join[dgt@i, dgt@j];
 cond2[i_, j_] := AllTrue[{{i, j}, {j, i}}, cond1];

```

```
 (* 頂点数 5のクリークができるまでグラフに頂点を追加する *)
 gr1 = Graph[{3 <-> 7}, VertexLabels -> Automatic];
 makeGraph[n_] := Module[{vertices},
   gr1 = VertexAdd[gr1, n];
   vertices =
    Select[VertexList@gr1, Divisible[n - #, 3] && cond2[n, #] &];
   If[vertices != {},
    gr1 = EdgeAdd[gr1, Flatten[{n <-> #} & /@ vertices]]]];
 i = PrimePi@11;
 While[FindClique[gr1, {5}] == {}, makeGraph@ Prime@i; i++];

 (* 和の最小値を与えるクリークの候補 *)
 curMin = First@MinimalBy[FindClique[gr1, {5}, All], Total];

 (* 素数のindex の最大値とクリークを表示 *)
 {i - 1, curMin}]

Out[]= {13.0876, {1051, {13, 5197, 5701, 6733, 8389}}}
```

## 3.2 最小性の検証

上で求めた $\{13, 5197, 5701, 6733, 8389\}$ の和 26033 よりも小さい和を与えるクリークがあるとすれば，その最大素数は $26033 - 3 - 7 - 11 - 13 = 25999$ 以下のはずです。

PrimePi$[25999] = 2860$ なので 2860 個目の素数までのグラフを調べれば十分です。和がもっと小さいクリークが見つかったら最小値を更新し，みつからなかったら上のコードで求めた 26033 が最小値です。

```
In[]:= Clear["Global`*"];
AbsoluteTiming[
 (* 2つの素数を辺で結ぶ条件 *)
 dgt[i_] := dgt[i] = IntegerDigits@i;
 cond1[{i_, j_}] := PrimeQ@FromDigits@Join[dgt@i, dgt@j];
 cond2[i_, j_] := AllTrue[{{i, j}, {j, i}}, cond1];

 (* 頂点数 5のクリークができるまでグラフに頂点を追加する *)
 gr1 = Graph[{3 <-> 7}, VertexLabels -> Automatic];
 makeGraph[n_] := Module[{vertices},
   gr1 = VertexAdd[gr1, n];
   vertices =
    Select[VertexList@gr1, Divisible[n - #, 3] && cond2[n, #] &];
   If[vertices != {},
```

```
    gr1 = EdgeAdd[gr1, Flatten[{n <-> #} & /@ vertices]]]];
i = PrimePi@11;
While[FindClique[gr1, {5}] == {}, makeGraph@ Prime@i; i++];

(* 和の最小値を与えるクリークの候補 *)
curMin = First@MinimalBy[FindClique[gr1, {5}, All], Total];

(* curMin よりも小さい和をあたえるクリークは存在するか *)
mx = Total@curMin - 3 - 7 - 11 - 13; (* 素数のインデックスの上限 *)
asc = GroupBy[Prime@Range[4, PrimePi@mx], Mod[#, 3] &];
pairs = Flatten[Subsets[#, {2}] & /@
    {Join[asc@1, {3}], Join[asc@2, {3}]}, 1];
makeEdge[{i_, j_}] :=
 If[cond2[i, j], UndirectedEdge[i, j], Nothing];

(* 検証用のグラフを作って計算 *)
gr2 = Graph[makeEdge /@ pairs, VertexLabels -> Automatic];
ans = First@MinimalBy[FindClique[gr2, {5}, All], Total];
ans = {Total@ans, ans == curMin}]

Out[]= {19.1159, {26033, True}}
```

結局，答えははじめに求めた値でした。そもそも検索範囲を広げても他に頂点数 5 のクリークは存在しませんでした。