

Project Euler 98. Anagramic Squares

hiragn

2024 年 12 月 27 日

1. 問題の概要

CARE という単語の各文字をそれぞれ 1, 2, 9, 6 に置き換えると平方数 $1296 = 36^2$ ができる。同じ置換によりアナグラムの RACE も平方数 $9216 = 96^2$ をつくる。CARE と RACE を平方アナグラム単語対と呼ぼう。先頭のゼロは許されず、異なる文字が同じ数字をもつこともないとする。

約 2,000 個の英単語を含むテキストファイル 0098_words.txt^a に含まれる平方アナグラム単語対を考える。回文となる単語はそれ自身のアナグラムとはみなさない。そのような対のメンバーから作られる最大の平方数は何か？

<https://projecteuler.net/problem=98>

^a https://projecteuler.net/project/resources/0098_words.txt

2. 解法

2.1 ファイルの中身を見てみよう

0098_words.txt に含まれる単語をアナグラムを作るグループに分けて抽出したら 42 組ありました。

```
{{"ACT", "CAT"}, {"ARISE", "RAISE"}, {"BOARD", "BROAD"}, {"CARE", "RACE"}, {"CENTRE", "RECENT"}, {"COURSE", "SOURCE"}, {"CREATION", "REACTION"}, {"CREDIT", "DIRECT"}, {"DANGER", "GARDEN"}, {"DEAL", "LEAD"}, {"DOG", "GOD"}, {"EARN", "NEAR"}, {"EARTH", "HEART"}, {"EAST", "SEAT"}, {"EAT", "TEA"}, {"EXCEPT", "EXPECT"}, {"FILE", "LIFE"}, {"FORM", "FROM"}, {"FORMER", "REFORM"}, {"HATE", "HEAT"}, {"HOW", "WHO"}, {"IGNORE", "REGION"}, {"INTRODUCE", "REDUCTION"}, {"ITEM", "TIME"},
```

```
{ "ITS", "SIT" }, { "LEAST", "STEAL" }, { "MALE", "MEAL" }, { "MEAN", "NAME" },
{ "NIGHT", "THING" }, { "NO", "ON" }, { "NOTE", "TONE" }, { "NOW", "OWN" },
{ "PHASE", "SHAPE" }, { "POST", "SPOT", "STOP" }, { "QUIET", "QUITE" },
{ "RATE", "TEAR" }, { "SHEET", "THESE" }, { "SHOUT", "SOUTH" },
{ "SHUT", "THUS" }, { "SIGN", "SING" }, { "SURE", "USER" }, { "THROW", "WORTH" }
```

- 文字列の長さは 2, 3, 4, 5, 6, 8, 9
- 3 単語の組もある
- 回文となる単語は含まれていない

2.2 置換のパターンに注目する

{ "CARE", "RACE" } と { 1296, 9216 } をどうやって結びつけるのか考えます。

文字列中の A, C, E, R は他の文字でも構わないので、「何番目のアルファベットか」のリストに直して置換のパターンを探します。

```
1 In[] := a = { "CARE", "RACE" };
2 LetterNumber /@ a
3 Out[] = {{3, 1, 18, 5}, {18, 1, 3, 5}}
4
5 In[] := FindPermutation @@ %
6 Out[] = Cycles[{{1, 3}}]
```

LetterNumber を ToCharacterCode や Characters に変えても結果は同じです。

```
1 In[] := FindPermutation @@ (ToCharacterCode /@ a)
2 Out[] = Cycles[{{1, 3}}]
3
4 In[] := FindPermutation @@ (Characters /@ a)
5 Out[] = Cycles[{{1, 3}}]
```

{ 1296, 9216 } も桁数字のリストに直すと、同じ置換のパターンだと分かります。

```
1 In[] := b = { 1296, 9216 };
2 IntegerDigits /@ b
3 Out[] = {{1, 2, 9, 6}, {9, 2, 1, 6}}
4
5 In[] := FindPermutation @@ %
6 Out[] = Cycles[{{1, 3}}]
```

これらを考えあわせると、4桁の平方数の中で「左から1番目の数字と3番目の数字を入れ替えても平方数であるもの」の最大値を求めれば

$$f(\{"CARE", "RACE"\}) = 9216$$

と関連付けられます。これを文字列の長さごとにやるとうまくいきそうです。

1. 長さ n の文字列の間の置換パターンを全部求める
2. n 桁の平方数の間の置換パターンを全部求める
3. これらに共通する平方数のペアを探して、大きい方の数を取り出す

3. 実装

上の方法でやってみたのですが、{"SHEET", "THESE"}のように同じ文字を複数個含む文字列があったり、桁数字にダブリがある平方数があったりしてうまくいきません。こういう文字や数字が置換で変化する場合としない場合を区別するために

{ 置換パターン, 文字の位置 }

を比較することにしました。

```
1 In[] :=
2 makePattern[pair_] := {FindPermutation @@ pair,
3   Values@PositionIndex@First@pair};
4 makePattern@{"S", "H", "E", "E", "T"}, {"T", "H", "E", "S", "E"}
5
6 Out[] = {Cycles[{{1, 4, 5}}], {{1}, {2}, {3, 4}, {5}}}
```

コード全体はこうなります。

```
1 In[] := Clear["Global`*"];
2 dat = First@Import["0098_words.txt", "CSV"];
3 RepeatedTiming[
4   anagrams = Select[GatherBy[dat, Sort@Characters@# &], Length@# > 1 &];
5
6 (* n 文字のアナグラムに対応する平方数の最大値 *)
7 calc[n_] :=
8   Module[{words, wordpairs, makePattern, patterns, squares,
9     squarepairs},
10    (* 文字列から置換パターンと文字の出現位置を抽出 *)
11    words = Characters /@ Select[anagrams, StringLength@First@# == n &];
12    wordpairs =
13      Flatten[{#, Reverse@#} & /@
```

```

14     Flatten[Subsets[#, {2}] & /@ words, 1], 1];
15     makePattern[pair_] := {FindPermutation @@ pair,
16       Values@PositionIndex@First@pair};
17     patterns = DeleteDuplicates[makePattern /@ wordpairs];
18
19     (* 文字列と同じパターンをもつ平方数の最大値を求める *)
20     squares =
21       Map[IntegerDigits, (Range[Ceiling@Sqrt[10^(n - 1)],
22         Floor@Sqrt[10^n - 1]])^2];
23     squarepairs =
24       Flatten[Subsets[#, {2}] & /@
25         Select[GatherBy[squares, Sort@# &], Length@# > 1 &], 1];
26     Max[FromDigits /@
27       Flatten[Select[squarepairs, MemberQ[patterns, makePattern@#] &],
28       1]]];
29
30     (* アナグラム文字列の長さ毎に計算 *)
31     ans = Max[calc /@ DeleteDuplicates[StringLength /@ Flatten@anagrams]]]
32
33 Out[] = {0.295317, 18769}

```
