Project Euler 54. Poker Hands

hiragn

2024 年 12 月 26 日

1. 問題の概要

ポーカーの役を弱い方から強い方へ並べると次のようになる。

- 役なし（ハイカード）：一番値が大きいカード
- ワン・ペア：同じ値のカードが 2 枚
- ツー・ペア：2 つの異なる値のペア
- スリーカード：同じ値のカードが 3 枚
- ストレート：5 枚の連続する値のカード
- フラッシュ：すべてのカードが同じスート（注：スートはダイヤ，ハート，クラブ，スペードのこと）
- フルハウス：スリーカードとペア
- フォーカード：同じ値のカードが 4 枚
- ストレートフラッシュ：ストレートかつフラッシュ
- ロイヤルフラッシュ：同じスートの 10, J, Q, K, A

カードの値は小さい方から順に 2〜10, J, Q, K, A である。データ中で 10 は T と表される。
2 人のプレイヤーが同じ役の場合は役を構成する中で値が最も大きいカードによってランクが決まる。たとえば 8 のペアは 5 のペアより強い。それでも同じランクの場合は，一番値が大きいカードによってランクが決まる。一番値が大きいカードが同じ場合には次に値が大きいカードが比べられ，以下同様に繰り返す。
`0054_poker.txt`[a] には 1000 個の手札の組が含まれている。各行は 10 枚のカードからなる。最初の 5 枚がプレイヤー 1 の手札であり，残りの 5 枚がプレイヤー 2 の手札である。以下のことを仮定してよい。

- 各プレイヤーの手札は特に決まった順に並んでいるわけではない

- 各勝負で勝敗は必ず決まる
- A, 2, 3, 4, 5 というストレートは考えなくてよい

1000 回中プレイヤー 1 が勝つのは何回か。

https://projecteuler.net/problem=54

[a] https://projecteuler.net/project/resources/0054_poker.txt

2. 解法

Xiangdong Zeng さんの解答[*1] を参考にしてパターンマッチングで解きました。競技プログラミングでいうところの解説 AC です。

```
In[]:= Clear["Global`*"];
dat = Partition[Characters /@ #, 5] & /@ Import["0054_poker.txt", "Table"];
RepeatedTiming[
 rules = Thread[Characters@"23456789TJQKA" -> Range[2, 14]];

 (*Royal Flush*)
 f[{10, s_}, {11, s_}, {12, s_}, {13, s_}, {14, s_}] := {9};

 (*Straight Flush*)
 f[{a_, s_}, {b_, s_}, {c_, s_}, {d_, s_}, {e_, s_}] :=
  {8, {e, d, c, b, a}} /;
   {a, b, c, d, e} == a + Range[0, 4];

 (*Four of a Kind*)
 f[{a_, _}, {b_, _}, {c_, _}, {d_, _}, {e_, _}] :=
  {7, Keys@ReverseSort@Counts[{a, b, c, d, e}]} /;
   Values@Sort@Counts[{a, b, c, d, e}] == {1, 4};

 (*Full House*)
 f[{a_, _}, {b_, _}, {c_, _}, {d_, _}, {e_, _}] :=
  {6, Keys@ReverseSort@Counts[{a, b, c, d, e}]} /;
   Values@Sort@Counts[{a, b, c, d, e}] == {2, 3};

 (*Flush*)
 f[{a_, s_}, {b_, s_}, {c_, s_}, {d_, s_}, {e_, s_}] :=
  {5, {e, d, c, b, a}};
```

[*1] https://stone-zeng.site/2020-08-22-euler-51-60

```
27
28  (*Straight*)
29  f[{a_, _}, {b_, _}, {c_, _}, {d_, _}, {e_, _}] :=
30   {4, {e, d, c, b, a}} /;
31    {a, b, c, d, e} == a + Range[0, 4];
32
33  (*Three of a Kind/Two Pairs/One Pair/High Card*)
34  f[{a_, _}, {b_, _}, {c_, _}, {d_, _}, {e_, _}] :=
35   With[{counts = ReverseSort@Counts[{a, b, c, d, e}]},
36    Switch[Values@counts,
37       (*Three of a Kind*)
38       {3, 1, 1}, {3, First[#], ReverseSort@Rest[#]},
39       (*Two Pairs*)
40       {2, 2, 1}, {2, ReverseSort@Most[#], Last[#]},
41       (*One Pair*)
42       {2, 1, 1, 1}, {1, First[#], ReverseSort@Rest[#]},
43       (*High Card*)
44       {1, 1, 1, 1, 1}, {0, {e, d, c, b, a}}] &@Keys[counts]];
45
46  (* 点数化して比較 *)
47  g[lst_] :=
48   FromDigits[PadRight[Flatten[f @@ Sort[lst /. rules]], 6], 15];
49  cond[lst1_, lst2_] := g@lst1 > g@lst2;
50  ans = Count[cond @@@ dat, True]]
51
52 Out[]= {0.0566867, 376}
```

まずはパターンマッチングで手札から次のようなリストを作ります。

{ 役の点数, 役に関係する数字（降順), 残りの数字（降順)}

役の点数はロイヤルフラッシュが 10 点で，以下 1 点ずつ減ってハイカード（役なし）が 0 点です。この部分は Xiangdong Zeng さんのコードをほぼそのまま使っています。

もとのコードではこのリストの引き算で勝敗の判定をしていましたが，私は点数化することにしました。

- PadRight で右に 0 を追加。リストの長さを 6 にそろえる
- リストを 15 進数の桁数字として FromDigits に渡して点数化する

スリーカード，ツーペア，ワンペア，ハイカードの例を載せておきます。

```
(* スリーカード *)
In[]:= g[{{13, "D"}, {5, "C"}, {5, "H"}, {5, "S"}, {6, "H"}}]
Out[]= 2576475

(* ツーペア *)
In[]:= g[{{13, "D"}, {5, "C"}, {5, "H"}, {6, "S"}, {6, "H"}}]
Out[]= 1842300

(* ワンペア *)
In[]:= g[{{13, "D"}, {5, "C"}, {5, "H"}, {6, "S"}, {7, "S"}}]
Out[]= 1058040

(* ハイカード *)
In[]:= g[{{13, "D"}, {5, "C"}, {8, "H"}, {6, "S"}, {7, "S"}}]
Out[]= 686795
```

この点数が高いほうが勝者です。パターンマッチングのいい練習になりました。

Stack Exchange の記事も参考になりました。

https://mathematica.stackexchange.com/questions/26179/classifying-poker-hands-by-pattern-matching