

一列に並べるのはハミルトンパスを作ること

hiragn

2024 年 12 月 11 日

「算数にチャレンジ!!」第 1257 回の問題^{*1}をハミルトンパス（すべての頂点を一度ずつ通るパス）の問題に言い換えて mathematica で解いた。

トモエさん、マサルさん、マサヒコくん、マリコさん、ツヨシくんの 5 人が横一列に並ぶ。次の条件をみたす並び方は何通りあるか。

- トモエさんはマサルさんの隣もマサヒコくんの隣も拒否する
- マリコさんはツヨシくんの隣は拒否する

<http://www.sansu.org/used-html/index1257.html>

1. ハミルトンパスを数える

5 人を頂点 1~5 であらわして、となりあう可能性のある 2 人の間に辺を張ると、この問題はグラフの問題に帰着できる。

- 1-2 間, 1-3 間, 4-5 間には辺を張らない。
- さらに始点として st, 終点として en という頂点を用意して、それぞれ頂点 1~5 との間に辺を張る。

こうして作ったグラフで st から en までのハミルトンパスの個数が答え。

頂点が少ないので答えは一瞬で出て、28 通りだった。

^{*1} この問題文は原題を適当に書き換えたもの。

```

1 In[]:= Clear["Global`*"];
2 RepeatedTiming[
3   g0 = Graph[Flatten[{st -> #, # -> en} & /@ Range@5]];
4   cond[i_, j_] :=
5     i != j && ! MemberQ[{{1, 2}, {1, 3}, {4, 5}}, Sort@{i, j}];
6   g1 = RelationGraph[cond, Range@5, DirectedEdges -> True];
7   g = GraphUnion[g0, g1];
8   ans = Length@FindPath[g, st, en, {6}, All]]
9
10 Out[] = {0.000327525, 28}

```

コード中の g_0 は始点、終点と 5 頂点を結ぶ辺だけのグラフで、 g_1 は 5 頂点間の辺だけのグラフ。これらを GraphUnion でまとめて 1 つのグラフ g にしている。

mathematica にはハミルトンパスを求める関数がないようなので、FindPath を使って st から en までの長さ 6 のパスをすべて求めた。

2. ハミルトンサイクルを数える

次にハミルトンサイクル（すべての頂点を一度ずつ通る閉路）を求める関数 FindHamiltonianCycle を使うコードを書いてみた。

上のコードで作ったグラフに en から st に向かう辺も加えて、ハミルトンサイクルの問題に言い換える。計算時間は少し増えるが、パスの長さ（＝辺の個数 \neq 頂点数）を指定しなくてすむ分、こちらの方がちょっと楽だと思う。

```

1 In[]:= Clear["Global`*"];
2 RepeatedTiming[
3   g0 = Graph[Flatten[{st -> #, # -> en} & /@ Range@5]];
4   cond[i_, j_] :=
5     i != j && ! MemberQ[{{1, 2}, {1, 3}, {4, 5}}, Sort@{i, j}];
6   g1 = RelationGraph[cond, Range@5, DirectedEdges -> True];
7   g = GraphUnion[{en -> st}, g0, g1];
8   ans = Length@FindHamiltonianCycle[g, All]]
9
10 Out[] = {0.000418399, 28}

```

3. 順列を数える

5 つのものの順列は $5! = 120$ 通りしかないので、愚直に全探索しても大した時間はかからない。

1. 1 から 5 までの数を並べ替えた順列を `Permutations` で作る
2. `Partition` を使って各順列を長さが 2 でオフセット値が 1 のサブリストに分割する
3. サブリストに $\{1, 2\}$, $\{1, 3\}$, $\{4, 5\}$ を含まない順列を `Select` してその個数を数える

実際にやってみたコードがこちら。グラフを使う解法の約 4 倍の時間がかかったが、それでも 0.001 秒程度で終わる。

```
1 In[] := Clear["Global`*"];
2 RepeatedTiming[
3   bad = {{1, 2}, {1, 3}, {4, 5}};
4   cond[lst_] := ContainsNone[Sort /@ Partition[lst, 2, 1], bad];
5   ans = Length@Select[Permutations@Range@5, cond]]
6
7 Out[] = {0.00135909, 28}
```
