パスカルの三角形とネットワーク

SHIMURA Masato

2017 年 6 月 15 日

目次

ProjectEuler の最初の方に最短パス数の数え上げ、と最多ウエイトのパスを見出す問題が入っている。何れもパスカルの三角形に関係し、マトリックス上に展開したネットワークを用いると簡潔に計算できるは、この手法の名は不知。

1　パスカルの３角形

- パスカルの三角形は次の簡潔なスクリプトで求められた。

```
      |:   (i.6)!/i.6
1 0  0  0 0 0
1 1  0  0 0 0
1 2  1  0 0 0
1 3  3  1 0 0
1 4  6  4 1 0
1 5 10 10 5 1
```

- 多項式の乗算は次で求めることができる

	a	b
a	a^2	ab
b	ab	b^2

$$a^2 \swarrow ab \swarrow$$
$$\swarrow ab \swarrow b^2$$

	a^2	$2ab$	b^2
a	a^3	$2a^2b$	ab^2
b	a^2b	$2ab^2$	b^3

$$a^3 \swarrow 2a^2b \swarrow ab^2 \swarrow$$
$$\swarrow a^2b \swarrow 2ab^2 \swarrow b^3$$

- 幾つかの J のイディオム
  pmul=: +//.@(*/) NB. 多項式の乗算

```
   1 2 pmul 1 2 pmul 1 2
1 6 12 8
```

$$(a+2b)(a+2b)(a+2b) = a^3 + 6a^2b + 12ab^2 + 8b^3$$

diag=: (<0 1)&|:@] NB. 斜めの取り出し

```
   diag i. 3 4
0 5 10
```

2 Jでのマトリクスの取出しと書き換え—準備体操

2.1 マトリクスの取り出し

- マトリクスの生成

```
 i. 3 4
0 1  2  3
4 5  6  7
8 9 10 11
```

- ブロックの取り出し (0,1,2 行 1,2 列)

```
      (<0 1 2;1 2) {i. 3 4
1  2
5  6
9 10
```

- ポイントの指定と取り出し (0 行 1 列、2 行 3 列)

```
      ((<0 1),<2 3){i.3 4
1 11
```

2.2 書き換え

- 書き換え　アメンド } を用いる

```
mat =. newdata (< point) } mat
```

```
   100 (<1 2)  } i. 3 4
0 1   2  3
4 5 100  7
8 9  10 11
```

3 パスカルの三角形とネットワーク

グラフ理論はマトリクスにのせると簡潔に計算できる。

3.1 経路の数え上げ

左上から右下への最短経路数を数え上げる。2 次で 口, 3 次で 田, そして 4 次で 囲である。

この数え上げ法は島田氏から教わった。
初期のセット。

```
   mk_mat 3
1 1 1 1
1 0 0 0
1 0 0 0
1 0 0 0
```

ここから 0 の個所を上と左とを足しながら移動する一個づつ

1 **1** 1 1
1 **2** 0 0
1 0 0 0
1 0 0 0

全てを足し上げると、ここでもパスカルの 3 角形が現れる。少し大きなものも手で計算できる
右上、左下の対角行列がパスカルの三角形を示している。

```
   calc_net   mk_mat 4
1 1  1  1  1
1 2  3  4  5
1 3  6 10 15
1 4 10 20 35
1 5 15 35 70
```

3.2 Script

スクリプトは多少煩瑣だが AI というほどのものではなく、アルゴリズムで一位に決定される。

- マトリクスの初期設定

```
mk_mat=: 3 : '(1,}.tmp0),>y#<-.tmp0=. 0,y#1'
NB. Usage: mk_mat 5
```

- 最短経路数の数え上げ。
  $\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$
  $b+d=e, c+e=f\cdots$ は、一個ずつ２重ループで計算し書き戻す。

```
calc_net=: 3 : 0
'raw column'=. $  mat=. y    NB. same $
for_ctr0. i. <: raw do.
 raw0=.ctr0{ mat            NB. top
  for_ctr1. i.<: column do.
    raw1=. (>: ctr0){ mat   NB. 2nd
    tmp=.+/((>:ctr1){ raw0),ctr1 {raw1
    mat=. tmp (<(>:ctr0),>: ctr1)}mat  NB. ammend
  end.
end.
)
```

4 宝拾い

ネットワークの各点に宝が置いてある。最短ステップで宝を拾ってゴールする。
次の方法がある。

- 最大の宝を拾うステップをもとめる
- 全経路を数え上げ、比較して最大ステップを求める

4.1 累計

- 累計を取る J のスクリプト
[*1]

```
     5?.5
4 1 3 2 0
   <\.&.|. 5?.5
+-+---+-----+-------+---------+
|4|1 4|3 1 4|2 3 1 4|0 2 3 1 4|
+-+---+-----+-------+---------+
   +/\.&.|. 5?.5
4 5 8 10 10
```

- 最初の行と列を累計する

```
   cum_mat i. 5 5
 0  1  3  6 10
 5  6  7  8  9
15 11 12 13 14
30 16 17 18 19
50 21 22 23 24
```

4.2

- 最大経路を求める。$d + d = c$ ではなく b, d の大きい方を取り、 (>./)、e に置かれている数字を加える。この方法で落ちたパスには最大値はないので計算が簡潔になる。

```
    (i. 5 5);(cum_mat i. 5 5); calc_net2 i. 5 5
+--------------+--------------+----------------+
| 0  1  2  3  4| 0  1  3  6 10| 0  1  3   6  10|
| 5  6  7  8  9| 5  6  7  8  9| 5 11 18  26  35|
|10 11 12 13 14|15 11 12 13 14|15 26 38  51  65|
|15 16 17 18 19|30 16 17 18 19|30 46 63  81 100|
|20 21 22 23 24|50 21 22 23 24|50 71 93 116 140|
+--------------+--------------+----------------+
  (i. 5 5)      (cum surround)    (calc)
```

- 5×5 では　右上左下の対角行列に　$2^4 = 1\,4\,6\,4\,1$ の対角となっている。
見やすくするため対角行列より右下を落とす

```
   (take_diag i. 5 5);(take_diag cum_mat i. 5 5);take_diag calc_net2 i. 5 5

+------------+-------------+--------------+
| 0  1  2 3 4| 0  1  3 6 10| 0  1  3  6 10|
```

[*1] もっと良いものがありそうだ

```
| 5  6  7 8 0| 5  6  7 8  0| 5 11 18 26  0|
|10 11 12 0 0|15 11 12 0  0|15 26 38  0  0|
|15 16  0 0 0|30 16  0 0  0|30 46  0  0  0|
|20  0  0 0 0|50  0  0 0  0|50  0  0  0  0|
+------------+-------------+--------------+
 i.5 5          cum surround   calc
```

- Script

```
take_diag=: 3 : 0
NB. drop out of pascal triangle
NB. usage: u i. 5 5
NR=. # tmp=.{ y
IND=. {@>|. >: i. NR
>IND {. L:0 tmp
)
```

- この手法で *projecteulerNo18* の解が得られる

パスカルの三角形をマトリクスにのせて簡単な計算で最大経路が見つかった。この方法の名は不知。

付録 A　全経路を求める方法

$2^2, 2^3, 2^4 \cdots$ の全経路を書き上げなければならない。巨大な疎マトリクスになる。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0		1	1											
1	1			1	1									
2	1					1	1							
3		1					1	1						
4		1							1	1				
5			1								1	1		
6			1										1	1
⋮														

このマトリクスにワーシャル・フロイド法は作用しなかった。

```
a=.   ,./>". L:0 readcsv '~temp/math/suuron/net_euler.csv'
 take_diag calc_net2 a
```

References