The Rank Operator and Dyadic Transpose

Richard Park

The Rank Operator

dyalog.tv/Webinar?v=IBct81IopRk

```
      {(f⍤     c)⍵}    ⍝ Monadic

     {α(f⍤  b  c)⍵}    ⍝ Dyadic

{α←⊢ ⋄ α(f⍤a b  c)⍵}    ⍝ Ambivalent
```

The Rank Operator

dyalog.tv/Webinar?v=IBct81IopRk

```
          {f⍤⍥     c⊢⍵}    ⍝ Monadic

        {⍺ f⍤⍥  b c⊢⍵}    ⍝ Dyadic

{⍺←⊢ ⋄ ⍺ f⍤⍥a b c⊢⍵}    ⍝ Ambivalent
```

The Rank Operator

dyalog.tv/Webinar?v=IBct81IopRk

```
      {g←f⍤0     c ⋄   g ⍵}

      {g←f⍤0   b c ⋄ ⍺ g ⍵}

{⍺←⊢ ⋄ g←f⍤0⍤a b c ⋄ ⍺ g ⍵}
```

The Rank Operator

dyalog.tv/Webinar?v=IBct81IopRk

```
      f⍤⍤     c       ⍝ Monadic
      f⍤⍤   b c       ⍝ Dyadic
      f⍤⍤a  b c       ⍝ Ambivalent
```

Advanced Use

dyalog.tv/Webinar/?v=5wW76XX0kqk

```
f⍤¯1 k   ⍝ Negative rank

⍤1 j⍤1 k ⍝ Multiple rank
```

Rank and Transpose

```
... ⍤⍤k ⊢ p⍉ ...
```

Merge Axes using Ravel

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```


```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      ,[2 3]3 2 4⍴⎕A
ABCDEFGH
IJKLMNOP
QRSTUVWX
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```


```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      ,⍤2⊢3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      ,⍤2⊢3 2 4⍴⎕A
ABCDEFGH

IJKLMNOP

QRSTUVWX
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      ,[1 2]3 2 4⍴⎕A
ABCD
EFGH
IJKL
MNOP
QRST
UVWX
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Dyadic Transpose

```
      3 2 4 ⍉ 4 3 2
2
  3
    1
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Dyadic Transpose

```
3 2 4 ⍉ 4 3 2
2 3 1
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      2 3 1⍉3 2 4⍴⎕A
AE
IM
QU

BF
...
...
SW

DH
LP
TX
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      2 3 1⍉3 2 4⍴⎕A
AE
IM
QU

BF
...
...
SW

DH
LP
TX
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      ,⍤2⊢2 3 1⍉3 2 4⍴⎕A
AEIMQU
BFJNRV
CGKOSW
DHLPTX
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

```
      ⍉,⍤2⊢2 3 1⍉3 2 4⍴⎕A
ABCD
EFGH
IJKL
MNOP
QRST
UVWX
```

```
      3 2 4⍴⎕A
ABCD
EFGH

IJKL
MNOP

QRST
UVWX
```

Merge Axes using Ravel

Dyalog Forums "Extended Inner Product"

forums.dyalog.com/viewtopic.php?f=13&t=1587

```
,[⍳⍺]⍵   ←→   {(1⌽⍳⍴⍴z)⍉z←,⍤⍺⊢((-⍺)⌽⍳⍴⍴⍵)⍉⍵}
```

Merge Axes using Ravel

Move axes to the end → p⍉ω

Merge N axes → ,⍤N

Move axes back → q⍉ω

Merge Axes

```
MergeAxes←{
     axes←⍳≠⍴⍵
     move←⍋(axes~⍺),⍺
     merged←,⍤(≠⍺)⊢move⍉⍵
     restore←((⍳≠⍴merged)~⊃⍺),⊃⍺
     restore⍉merged
}
```

Last Axis Primitives

```
      ↓2 3 2⍴⍳12
```

1 2	3 4	5 6
7 8	9 10	11 12

```
      □←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
      ↓[1]2 3 2⍴⍳12
```

1 7	2 8
3 9	4 10
5 11	6 12

```
      □←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
      ↓[1]2 3 2⍴⍳12
```

1 7	2 8
3 9	4 10
5 11	6 12

```
      □←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
      ↓[2]2 3 2⍴⍳12
```

1 3 5	2 4 6
7 9 11	8 10 12

```
      ⎕←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
      ↓[2]2 3 2⍴⍳12
```

1 3 5	2 4 6
7 9 11	8 10 12

```
      ⎕←2 3 2⍴⍳12
1  2
3  4
5  6

7  8
9 10
11 12
```

Last Axis Primitives

```
      ↓[3]2 3 2⍴⍳12
```

1 2	3 4	5 6
7 8	9 10	11 12

```
      ⎕←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
      ↓2 3 2⍴⍳12
```

1 2	3 4	5 6
7 8	9 10	11 12

```
      □←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
      (↓⍤1)2 3 2⍴⍳12
```

1 2	3 4	5 6
7 8	9 10	11 12

```
      ⎕←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
      (↓⍤2)2 3 2⍴⍳12
```

1 2	3 4	5 6
7 8	9 10	11 12

```
      ⎕←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
      (↓⍤3)2 3 2⍴⍳12
```

1 2	3 4	5 6
7 8	9 10	11 12

```
      ⎕←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Last Axis Primitives

```
(⊂⍤1)2 3 2⍴⍳12
```

1 2	3 4	5 6
7 8	9 10	11 12

```
      ⎕←2 3 2⍴⍳12
 1  2
 3  4
 5  6

 7  8
 9 10
11 12
```

Dyalog Webinars: Language Features of Dyalog version 18.0 in Depth - Part 5

dyalog.tv/Webinar/?v=HU8jebyXKqc

Split: Rank + Transpose

```
      ↓[1]2 3 4⍴⎕A
```

AM	BN	CO	DP
EQ	FR	GS	HT
IU	JV	KW	LX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Split: Rank + Transpose

```
(⊂⍤1)3 1 2⍉2 3 4⍴⎕A
```

AM	BN	CO	DP
EQ	FR	GS	HT
IU	JV	KW	LX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Split: Rank + Transpose

```
      ↓[2]2 3 4⍴⎕A
```

AEI	BFJ	CGK	DHL
MQU	NRV	OSW	PTX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Split: Rank + Transpose

```
(⊂⍤1)1 3 2⍉2 3 4⍴⎕A
```

AEI	BFJ	CGK	DHL
MQU	NRV	OSW	PTX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Split: Rank + Transpose

```
      ↓[3]2 3 4⍴⎕A
```

ABCD	EFGH	IJKL
MNOP	QRST	UVWX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Split: Rank + Transpose

```
      ↓2 3 4⍴⎕A
```

ABCD	EFGH	IJKL
MNOP	QRST	UVWX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Split: Rank + Transpose

```
      (⊂⍤1)2 3 4⍴⎕A
```

ABCD	EFGH	IJKL
MNOP	QRST	UVWX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂2 3 4⍴⎕A
```

```
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂[1]2 3 4⍴⎕A    ⍝ ↓[1]
```

AM	BN	CO	DP
EQ	FR	GS	HT
IU	JV	KW	LX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
⊂[2]2 3 4⍴⍳⍬A    ⍉ ↓[2]
```

AEI	BFJ	CGK	DHL
MQU	NRV	OSW	PTX

```
      ⍳⍬←2 3 4⍴⍳⍬A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂[3]2 3 4⍴⎕A    ⍝ ↓[3]
```

ABCD	EFGH	IJKL
MNOP	QRST	UVWX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂[1 2]2 3 4⍴⍳9
```

AEI MQU	BFJ NRV	CGK OSW	DHL PTX

```
      ⍳←2 3 4⍴⍳9
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
(⊂⍤2)2 3 1⍴2 3 4⍴⎕A
```

AEI MQU	BFJ NRV	CGK OSW	DHL PTX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂[2 3]2 3 4⍴⎕A
```

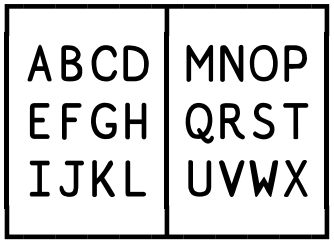
ABCD EFGH IJKL	MNOP QRST UVWX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      (⊂⍤2)2 3 4⍴'ABCDEFGHIJKLMNOPQRSTUVWX'
```




```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂[1 3]2 3 4⍴⎕A
```

ABCD MNOP	EFGH QRST	IJKL UVWX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      (⊂⍤2)2 1 3⍴2 3 4⍴⎕A
```

ABCD MNOP	EFGH QRST	IJKL UVWX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂[3 1]2 3 4⍴⎕A
```

AM	EQ	IU
BN	FR	JV
CO	GS	KW
DP	HT	LX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      (⊂⍤2)3 1 2⍉2 3 4⍴⎕A
```

AM	EQ	IU
BN	FR	JV
CO	GS	KW
DP	HT	LX

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂[1 2 3]2 3 4⍴⎕A
```

```
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂2 3 4⍴⍳12
```

```
┌────┐
│ABCD│
│EFGH│
│IJKL│
│    │
│MNOP│
│QRST│
│UVWX│
└────┘
```

```
      ⍳←2 3 4⍴⍳12
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂[2 3 1]2 3 4⍴⎕A
```

```
AM
BN
CO
DP

EQ
FR
GS
HT

IU
JV
KW
LX
```

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂(⍋2 3 1)⌽2 3 4⍴⎕A
```

```
AM
BN
CO
DP

EQ
FR
GS
HT

IU
JV
KW
LX
```

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      ⊂3 1 2⍉2 3 4⍴⎕A
```

```
┌→──┐
↓AM │
↓BN │
↓CO │
↓DP │
│   │
↓EQ │
↓FR │
↓GS │
↓HT │
│   │
↓IU │
↓JV │
↓KW │
↓LX │
└───┘
```

```
      ⎕←2 3 4⍴⎕A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose

```
      )copy dfns pmat
      pmat¨⍳3
```

1	1	2	1	2	3
	2	1	1	3	2
			2	1	3
			2	3	1
			3	1	2
			3	2	1

```
      □←2 3 4⍴□A
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

Enclose Axes

```
EncloseAxes←{
     axes←⍳≢⍴⍵
     move←⍋(axes~⍺),⍺
     ⊂⍤(≢⍺)⊢move⍉⍵
}
```

Merge Axes

```
MergeAxes←{
     axes←⍳≢⍴⍵
     move←⍋(axes~⍺),⍺
     merged←,⍤(≢⍺)⊢move⍉⍵
     restore←((⍳≢⍴merged)~⊃⍺),⊃⍺
     restore⍉merged
}
```

General Purpose Array Algorithm

1. Identify axes of interest
2. Move axes of interest to end (⍉)
3. Apply leading-axis function with rank (⍤)
4. Move trailing axes back (⍉)

General Purpose Array Algorithm (slow)

1. Identify axes of interest
2. Enclose axes of interest (⊂[ ])
3. Apply function with each (¨)
4. Mix (↑)

General Purpose Array Algorithm

1. Identify axes of interest
2. Move axes of interest to end (⍉)
3. Apply leading-axis function with rank (⍤)
4. Move trailing axes back (⍉)
5. Profit!

Array Structure Manipulation

```
⊂⍤⍥⊃   ⍝ Increase nesting
 ↑     ⍝ Decrease nesting

,⍤⍥⊃   ⍝ Merge data

 ⍉     ⍝ Reorder axes
```

Summary: The Rank Operator

```
f⍤c          ⍝ Monadic
f⍤b c        ⍝ Dyadic
f⍤a b c      ⍝ Ambivalent
f⍤¯k         ⍝ Negative
f⍤k⍤j        ⍝ Multiple

q⍤f⍤n⊢p⍤      ⍝ Reorder axes
```

Next Week

British APL Association Open Session

britishaplassociation.org/webinar-schedule-2020

October 8th 15:00 UTC

Dyalog '20 Online

Monday 9th - Tuesday 10th November

Register: dyalog.com/user-meetings/dyalog20.htm