Progressive Set Functions

Adám Brudzewsky

Links

TryAPL lesson

tinyurl.com/vx67y6b

APL Cultivation lesson

tinyurl.com/to2na6w

Anatomy of an Idiom, Bob Smith

tinyurl.com/t2f5h9h

Index of (⍳)

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2 1 1 2 1 1 4
```

Index of (⍳)

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2 1 1 2 1 1 4
```

Index of (⍳)

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2 1 1 2 1 1 4
```

Index of (⍳)

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2 1 1 2 1 1 4
```

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
```

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2
```

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2 1
```

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2 1 3
```

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2 1 3 5
```

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

```
a b a c b a
b a a b a a c
2 1 3 5 6
```

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

a	b	a	c	b	a	•
b	a	a	b	a	a	c
2	1	3	5	6	7	

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

a	b	a	c	b	a	·
b	a	a	b	a	a	c
2	1	3	5	6	7	4

Index of (⍳) *without replacement*

```
    L ← 'a1' 'b1' 'a2' 'c1' 'b2' 'a3'
    R ← 'b1' 'a1' 'a2' 'b2' 'a3' 'a4' 'c1'
      L⍳R
2 1 3 5 6 7 4
```

Labelling

```
        ↑ L (L⍳L)
a b a c b a
1 2 1 4 2 1
        ↑ R (R⍳R)
b a a b a a c
1 2 2 1 2 2 7
```

Labelling (the ranking function)

```
      ↑ L (L⍳L) (⍋⍋L⍳L)
a b a c b a
1 2 1 4 2 1
1 4 2 6 5 3
```

Almost there…

Almost there…

```
      ↑ L (⍋⍋L⍳L) R (⍋⍋R⍳R)
a b a c b a
1 4 2 6 5 3
a a a b c b
1 2 3 4 6 5
      (⍋⍋L⍳L)⍳(⍋⍋R⍳R)
1 3 6 2 4 5
```

Almost there…

1. The arrays contain the same unique major cells
2. The arrays must have equally many of each unique major cell
3. The unique major cells occur in the same order

```
      L ← 'abac' ⋄ R ← 'pqrs'
      ↑ L (⍋⍋L⍳L) R (⍋⍋R⍳R)
a b a c
1 3 2 4
p q r s
1 2 3 4
```

Almost there…

1. The arrays share the same set of major cells
2. The arrays must have equally many of each unique major cell
3. The unique major cells occur in the same order

```
      L ← 'abac' ⋄ R ← 'abab'
      ↑ L (⍋⍋L⍳L) R (⍋⍋R⍳R)
a b a c
1 3 2 4
a b a b
1 3 2 4
```

Almost there...

1. The arrays share the same set of major cells
2. The arrays must have equally many of each unique major cell
3. The unique major cells occur in the same order

```
      L ← 'abc' ⋄ R ← 'cba'
      ↑ L (⍋⍋L⍳L) R (⍋⍋R⍳R)
a b c
1 2 3
c b a
1 2 3
```

Almost there...

```
      ↑ L (⍋⍋L⍳L) R (⍋⍋R⍳R)
a b a c b a
1 4 2 6 5 3
a a a b c b
1 2 3 4 6 5
      (⍋⍋L⍳L)⍳(⍋⍋R⍳R)
1 3 6 2 4 5
```

Almost there…

1. The arrays share the same set of major cells
2. The arrays must have equally many of each unique major cell
3. The unique major cells occur in the same order

```
      L ← 'abc' ⋄ R ← 'cba'
      ↑ L (⍋⍋LⳡL) R (⍋⍋RⳡR)
a b c
1 2 3
c b a
1 2 3
```

Almost there…

1. The arrays share the same set of major cells
2. The arrays must have equally many of each unique major cell

```
      L ← 'abc' ⋄ R ← 'cba'
      ↑ L (⍋⍋L⍳L) R (⍋⍋L⍳R)
a b c
1 2 3
c b a
3 2 1
```

Almost there…

1. The arrays share the same set of major cells
2. The arrays must have equally many of each unique major cell

```
      L ← 'abc' ⋄ R ← 'cba'
      ↑ L (⍋⍋L⍳L) R (⍋⍋L⍳R)
a b c
1 2 3
c b a
3 2 1
```

Almost there...

```
          ↑ (L⍪R) (⍋⍋L⍳L⍪R) (R⍪L) (⍋⍋L⍳R⍪L)
a b a  c b a  b a  a  b  a  a  c
1 8 2 12 9 3 10 4  5 11  6  7 13
b a a  b a a  c a  b  a  c  b  a
8 1 2  9 3 4 12 5 10  6 13 11  7
```

Almost there...

```
          ↑ (L⍪R) (⍋⍋L⍳L⍪R) (R⍪L) (⍋⍋L⍳R⍪L)
a b a  c b a  b a  a  b  a  a  c
1 8 2 12 9 3 10 4  5 11  6  7 13
b a a  b a a  c a  b  a  c  b  a
8 1 2  9 3 4 12 5 10  6 13 11  7
          (⍋⍋L⍳L⍪R)⍳(⍋⍋L⍳R⍪L)
2 1 3 5 6 8 4 9 7 11 13 10 12
```

Almost there...

```
          ↑ (L⍪R) (⍋⍋L⍳L⍪R) (R⍪L) (⍋⍋L⍳R⍪L)
 a b a  c b a  b a  a  b  a  a  c
 1 8 2 12 9 3 10 4  5 11  6  7 13
 b a a  b a a  c a  b  a  c  b  a
 8 1 2  9 3 4 12 5 10  6 13 11  7
          (⍋⍋L⍳L⍪R)⍳(≢R)↑(⍋⍋L⍳R⍪L)
 2 1 3 5 6 8 4
```

Huzzah!

```
          ↑ (L⍪R) (⍋⍋L⍳L⍪R) (R⍪L) (⍋⍋L⍳R⍪L)
a b a  c b a  b a  a  b  a  a  c
1 8 2 12 9 3 10 4  5 11  6  7 13
b a a  b a a  c a  b  a  c  b  a
8 1 2  9 3 4 12 5 10  6 13 11  7
((≢L)↑(⍋⍋L⍳L⍪R))⍳(≢R)↑(⍋⍋L⍳R⍪L)
2 1 3 5 6 7 4
```

Index of (⍳) *without replacement*

```
L ← 'abacba' ⋄ R ← 'baabaac' ⋄ ⎕←↑ L R (L⍳R)
```

a	b	a	c	b	a	·
b	a	a	b	a	a	c
2	1	3	5	6	7	4

Progressive dyadic epsilon

```
      ⊢pdi←L{((≢⍺)↑(⍋⍋⍺⍳⍺⍪⍵))⍳(≢⍵)↑(⍋⍋⍺⍳⍵⍪⍺)}R
2 1 3 5 6 7 4
      pdi≤≢L
1 1 1 1 1 0 1
```

Progressive dyadic epsilon

```
      L{((≢⍺)↑(⍋⍋⍺⍳⍺⍪⍵))⍳(≢⍵)↑(⍋⍋⍺⍳⍵⍪⍺)}R
2 1 3 5 6 7 4
      R{((≢⍺)↑(⍋⍋⍺⍳⍺⍪⍵))∊(≢⍵)↑(⍋⍋⍺⍳⍵⍪⍺)}L
1 1 1 1 1 0 1
```

Cheeky operator

```
      _WR←{((≢⍺)↑⍋⍋⍺⍳⍺⍪⍵)⍺⍺((≢⍵)↑⍋⍋⍺⍳⍵⍪⍺)}
      L ⍳_WR R
2 1 3 5 6 7 4
      R ∊_WR L
1 1 1 1 1 0 1
```

Come fly with me

F	B	P	E
First Class	Business	Premium Economy	Economy

Come fly with me

Seats		Passengers
F	FF	FFFFFFFFPPEEEEEEEEEEEEEEEEEEEEE
F	FF	
PP	PP	
PP	PP	
PP	PP	
EE	EE	
PP	PP	
EE	EE	
EE	EE	
EE	EE	

Come fly with me

<u>Seats</u>

```
⍤   ⍤⍤
⍤   ⍤⍤
⍤⍤  PP
PP  PP
PP  PP
⍤⍤  ⍤⍤

PP  PP
⍤⍤  ⍤⍤
⍤⍤  ⍤⍤
⍤⍤  ⍤⍤
```

<u>Passengers</u>

```
FFFFFFFFPPEEEEEEEEEEEEEEEEEEEEE
```

```
seats{'⍤'@(∊_WR∘⍵)⍺}passengers
```

Come fly with me

<u>Seats</u>

```
⍤    ⍤⍤
⍤    ⍤⍤
⍤⍤   PP
PP   PP
PP   PP
⍤⍤   ⍤⍤

PP   PP
⍤⍤   ⍤⍤
⍤⍤   ⍤⍤
⍤⍤   ⍤⍤
```

<u>Passengers</u>

```
FFFFFFFFPPEEEEEEEEEEEEEEEEEEEEE

      ≢passengers
31
      passengers(+/∊⍨WR)seats
24
```

Without Replacement operator

```
_WR←{((≢⍺)↑⍋⍋⍺⍳⍺⍪⍵)⍺⍺((≢⍵)↑⍋⍋⍺⍳⍵⍪⍺)}
```

lookup without replacement

```
PDI←⍳_WR
```

element without replacement

```
PDE←∊_WR
```

Links

TryAPL lesson

tinyurl.com/vx67y6b

APL Cultivation lesson

tinyurl.com/to2na6w

Anatomy of an Idiom, Bob Smith

tinyurl.com/t2f5h9h

Next webinar

Thursday
16th April
15:00 UTC