How I Won The APL Problem Solving Competition (Professional Entrant)

Alexander Block
(Viridium Insurance Group)

Introduction

A few words about myself and how I got into contact with APL.

⍺⌊∈×⍺∩⊃∈⌈ ⍰⌊○⊂⌉<

Introduction

A few words about myself and how I got into contact with APL.

⍺⌊∈×⍺∩⊃∈⌈ ⍰⌊○⊂⌉<

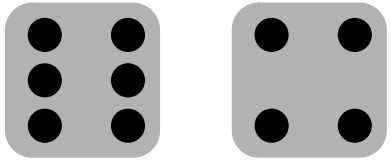
Now onwards to the problems...

https://github.com/Alexander-Block/dyalog-competition-2023

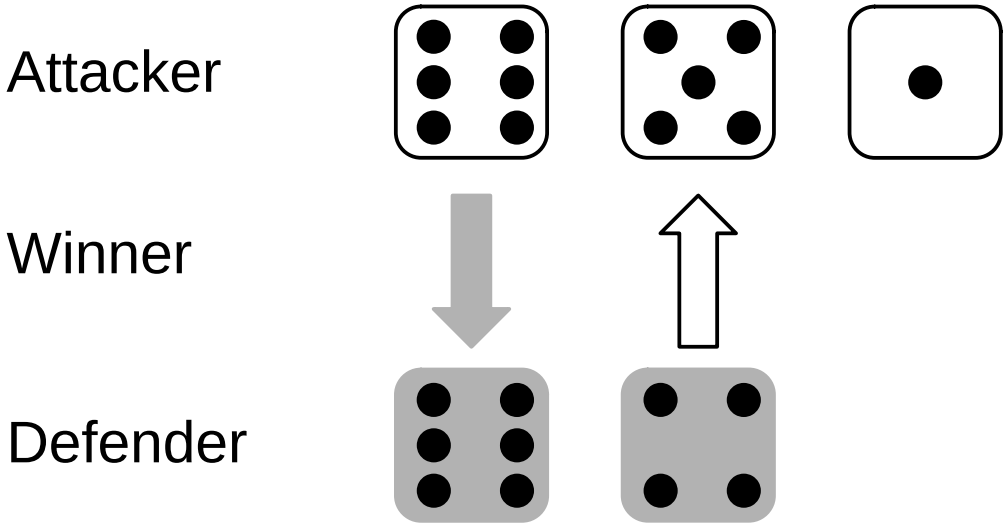
Risky Business

Attacker

Defender



Risky Business



Risky Business

```
1 1 ≡ 6 5 1 risk 6 4
```

Risky Business

```
1 1 ≡ 6 5 1 risk 6 4

2 1 ≡ 6 6 4 2 1 risk 6 5 5
```

Risky Business

```
1 1 ≡ 6 5 1 risk 6 4

2 1 ≡ 6 6 4 2 1 risk 6 5 5

0 1 ≡ 6 riskö, 5
```

Risky Business

```
1 1 ≡ 6 5 1 risk 6 4

2 1 ≡ 6 6 4 2 1 risk 6 5 5

0 1 ≡ 6 riskö, 5

1 2 ≡ 4 0 ¯1 risk 3 1 ¯2
```

Risky Business

```
risk←{
      len←(≢⍺)⌊(≢⍵)
      def_lost←+/(len↑⍵)<(len↑⍺)
      (len-def_lost)def_lost
}
```

Risky Business

```
risk←{
      len←⍺⌊⍤≠⍵
      def_lost←+/(len↑⍵)<(len↑⍺)
      (len-def_lost)def_lost
}
```

Risky Business

```
risk←{
      len←⍺⌊⍤≠⍵
      def_lost←+/>⍤(len∘↑)
      len(-,⊢)def_lost
}
```

Not my actual solution

Array Partitioning

```
res ← spec partition array
```

```
spec ↔ (shape movement first)
```

Array Partitioning

```
res ← spec partition array

spec ↔ (shape movement first)
```

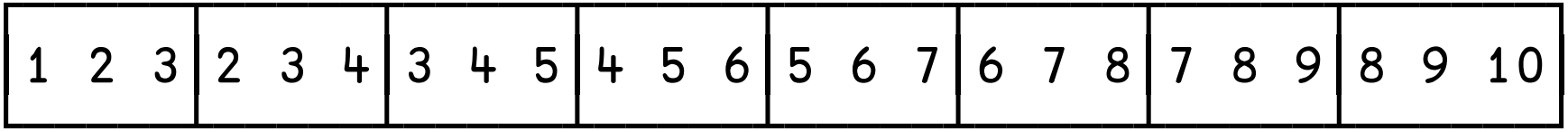

may be
of depth
0 or 1

optional

optional

Array Partitioning

```
3 partition ⍳10
```



Array Partitioning

```
      3 partition ⍳10
```

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

```
      (,3) (,2) 4 partition ⍳10
```

4 5 6	6 7 8	8 9 10

Array Partitioning

```
(2 2) (2 3) (2 1) partition 5 5ρ⎕A
```

FG KL	IJ NO	PQ UV	ST XY

```
ρ⍤⍤(7 partition ⍳6) (⊃7 partition ⍳6)
```

0	7

Array Partitioning

Step 1: Sort out partition specification

```
partition←{
    spec←{(0<≢¨⍵)/⍵},⊆⍺
    mov_def←(≢1⊃spec)⍴1
    first_def←(≢⍴⍵)⍴1
    (shape movement first)←⌽¯3↑first_def mov_def,⌽spec
    (shape movement)←{⌽1+(≢⍴array)↑¯1+⌽⍵}¨shape movement
    ...
}
```

Array Partitioning

Step 2: Use `first` parameter

```
partition←{
     ...
     array←(¯1+first)↓⍵
     ...
}
```

Array Partitioning

Step 3: Handle case with `shape` too large

```
partition←{
     ...
     array←(¯1+first)↓⍵
     ∨/shape>⍴array: 0⍴⊂shape⍴array
     ...
}
```

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

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

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

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

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

1	2	**1**	**2**	**3**	2	3	4	3	4	5	4	5	6	5	6	7	6	7	8	7	8	9	8	9	10	9	10

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

1	2	1	2	3	**2**	**3**	**4**	3	4	5	4	5	6	5	6	7	6	7	8	7	8	9	8	9	10	9	10

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

1	2	1	2	3	2	3	4	**3**	**4**	**5**	4	5	6	5	6	7	6	7	8	7	8	9	8	9	10	9	10

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

1	2	1	2	3	2	3	4	3	4	5	**4**	**5**	**6**	5	6	7	6	7	8	7	8	9	8	9	10	9	10

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

1 2	1 2 3	2 3 4	3 4 5	4 5 6	**5 6 7**	6 7 8	7 8 9	8 9 10	9 10

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7	**6**	**7**	**8**	7	8	9	8	9	10	9	10

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

1 2	1 2 3	2 3 4	3 4 5	4 5 6	5 6 7	6 7 8	**7 8 9**	8 9 10	9 10

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

```
( 1 2 3 4 5 6 7 8 9 10 )
```

1 2	1 2 3	2 3 4	3 4 5	4 5 6	5 6 7	6 7 8	7 8 9	**8 9 10**	9 10

Array Partitioning

```
{⊂⍺↓⍵}⌸3⊢⍳10
```

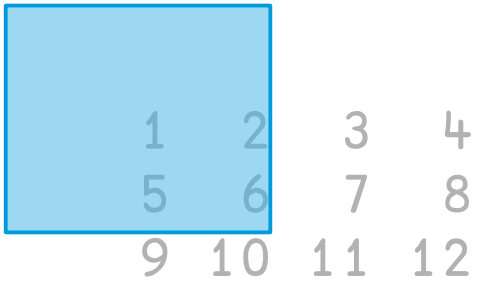
```
( 1 2 3 4 5 6 7 8 9 10 )
```

1 2	1 2 3	2 3 4	3 4 5	4 5 6	5 6 7	6 7 8	7 8 9	8 9 10	**9 10**

Array Partitioning

```
{⊂α↓ω}⌺3 3⊢3 4⍴⍳12
```

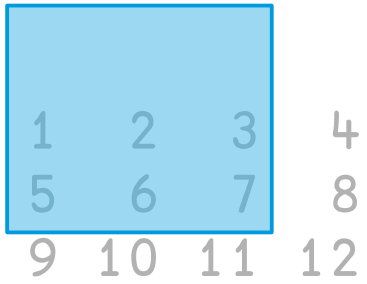
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂α↓ω}⌸3 3⊢3 4⍴⍳12
```

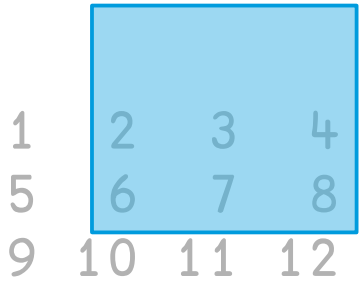
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂α↓ω}⌺3 3⊢3 4⍴⍳12
```

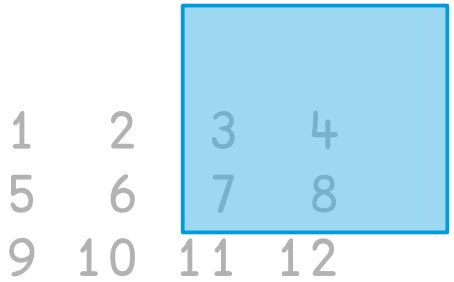
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂α↓ω}⌸3 3⊢3 4⍴⍳12
```

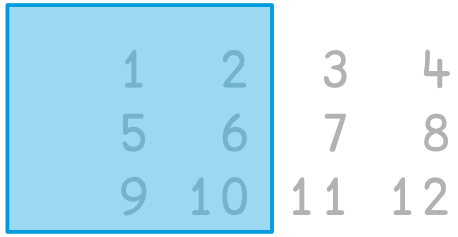
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂α↓ω}⌺3 3⊢3 4⍴⍳12
```

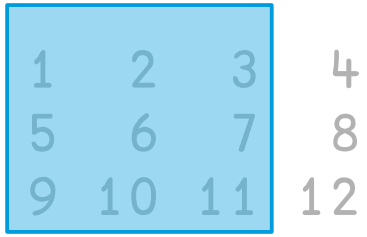
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂α↓ω}⌸3 3⊢3 4⍴⍳12
```

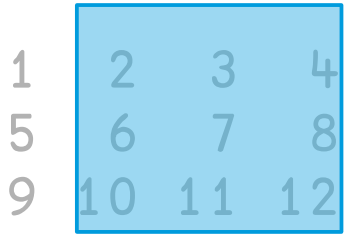
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂α↓ω}⌸3 3⊢3 4⍴⍳12
```

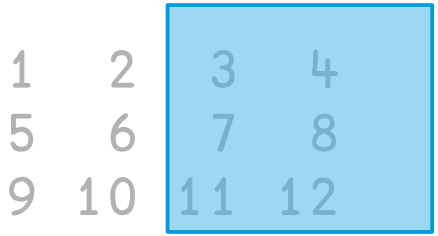
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂⍺↓⍵}⌸3 3⊢3 4⍴⍳12
```

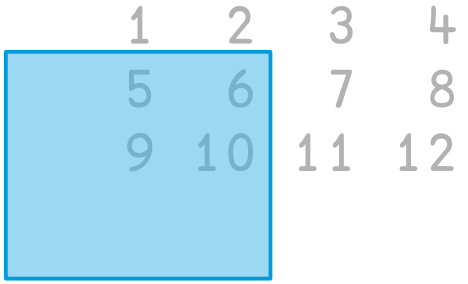
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂⍺↓⍵}⌸3 3⊢3 4⍴⍳12
```

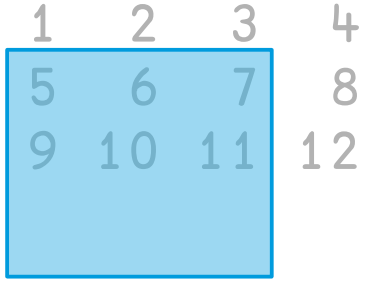
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂⍺↓⍵}⌸3 3⊢3 4⍴⍳12
```

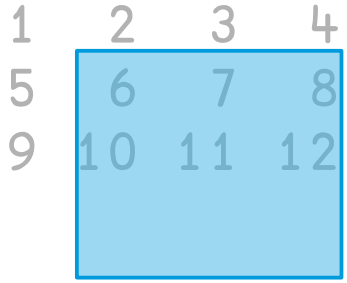
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂⍺↓⍵}⌸3 3⊢3 4⍴⍳12
```

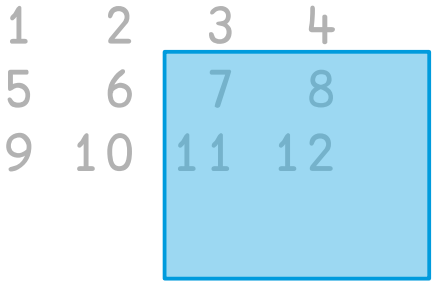
1 2 5 6	1 2 3 5 6 7	2 3 4 6 7 8	3 4 7 8
1 2 5 6 9 10	1 2 3 5 6 7 9 10 11	2 3 4 6 7 8 10 11 12	3 4 7 8 11 12
5 6 9 10	5 6 7 9 10 11	6 7 8 10 11 12	7 8 11 12



Array Partitioning

```
{⊂α↓ω}⌺3 3⊢3 4⍴⍳12
```

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



Array Partitioning

Step 4: Partition using stencil

```
partition←{
    ...
    partition_with_overhang←({⊂⍺↓⍵}⌺shape)array
    i_full←(⍳(⊂shape)≡¨⍴¨partition_with_overhang)
    ...
}
```

Array Partitioning

Step 5: Determine and keep relevant part of partition

```
partition←{
    ...
    partition_with_overhang←({⊂⍺↓⍵}⌺shape)array
    i_full←(⍳(⊂shape)≡¨⍴¨partition_with_overhang)
    i_start←⊃⌊/i_full
    i_keep←(∧/¨0=(⊂movement)|(i_full-⊂i_start))/i_full
    ⊃¨i_keep[]¨⊂partition_with_overhang
}
```

Conclusion

Thank you!