Processing Lists of Words

Rich Park

Asher Harvey-Smith

Adám Brudzewsky

Learning Objectives

- Understand the difference between flat and nested arrays in terms of the array model as implemented in Dyalog
- Use flat array techniques to perform operations on delimited lists and character matrices
- Understand the trade-offs between performance and complexity when using flat or nested arrays

Plan

- Review of key techniques used in nested solutions
- Introduce alternative list formats
- Participants in breakout rooms attempt [subset of] problems using alternative formats

The Array Model

In simple terms, simple arrays in memory:

[shape…], [elements in ravel order…]

The Array Model

In simple terms, simple arrays in memory:

2 3 4 ABCDEFGHIJKLMNOPQRSTUVWX

The Array Model

In simple terms, simple arrays in memory:

2 3 4 ABCDEFGHIJKLMNOPQRSTUVWX

```
ABCD
EFGH
IJKL

MNOP
QRST
UVWX
```

The Array Model

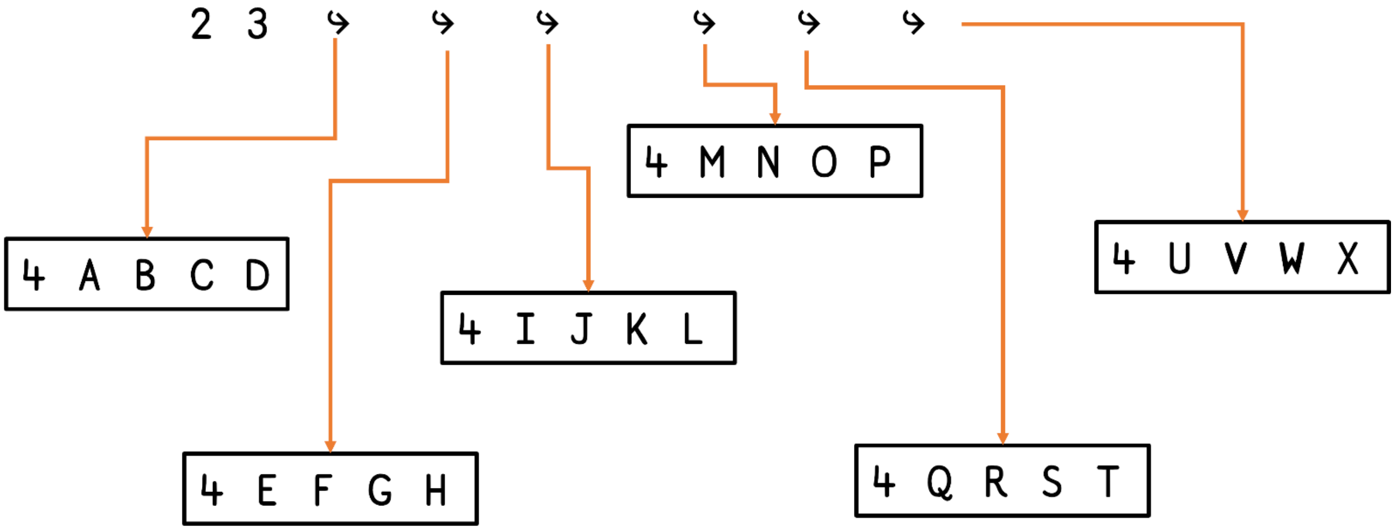
In simple terms, **nested arrays** in memory:

```
┌────┬────┬────┐
│ABCD│EFGH│IJKL│
├────┼────┼────┤
│MNOP│QRST│UVWX│
└────┴────┴────┘
```

The Array Model

```
┌────┬────┬────┐
│ABCD│EFGH│IJKL│
├────┼────┼────┤
│MNOP│QRST│UVWX│
└────┴────┴────┘
```

In simple terms, **nested arrays** in memory:



Word List Formats

```
      words
┌──────┬───┬────────┐
│MONKEY│DOG│ELEPHANT│
└──────┴───┴────────┘
```

Word List Formats

```
      wmat←↑words
MONKEY
DOG
ELEPHANT
```

Word List Formats

```
      wmat←↑words
MONKEY□□
DOG□□□□□
ELEPHANT
```

Word List Formats

```
      wmat←∊,∘';'¨words
MONKEY;DOG;ELEPHANT;
```

- Similar to newline characters in text files

Word List Formats

```
      ]display ∊,∘(⎕UCS 10)¨'MONKEY' 'DOG' 'ELEPHANT'
┌→───────┐
│MONKEY  │
│DOG     │
│ELEPHANT│
│        │
└────────┘
```

End of each line is a newline character

```
      ⎕SIZE'words'
2203376
      ⎕SIZE'wmat'
863824
      ⎕SIZE'wdel'
38584
```

Benefits of Flat Arrays

- Smaller size

  Use less memory

- Friendly traversal patterns

  No chasing pointers

Drawbacks of Flat Arrays

Flat array techniques can be more complicated than their nested counterparts.

Timing

Aim for 2× faster

```
        ]runtime -c "expr1" "expr2"
-50%      2× faster
+100%     2× slower
```

Timing

Try this now

```
]runtime -c "⎕DL 0.3" "⎕DL 0.6“
```

```
  ⎕dl 0.3 → 3.1E¯1 |    0% ⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕
* ⎕dl 0.6 → 6.1E¯1 | +96% ⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕⎕
```

Aims

All problems can be solved by converting to another format.

We want to find optimal solutions for a particular format.

```
      +/'A'∊¨words
21313
      +/'A'∊¨↓wmat
21313
      +/'A'∊¨';'(≠⊆⊢)wdel
21313
```

Activity

https://dyalog-training.github.io/2025-ListsWorkshop/

Day 1