

March 13, 2025

Announcements

- HW graded
- Talk uploads

Goals

- Grab bag of accidents

Review

$$S = \{(i, j) \mid 0 \leq i < N \wedge 0 \leq j < M\}$$

\uparrow $M=0 \Rightarrow S = \emptyset$

$$\prod_{i \in S} \sum (i) = 0$$

Arrays



Why are **arrays** the dominant data structure in high-performance code?

boxes!

Any comments on C's arrays?

float s[] [7],

real * s s(m, n)

Arrays vs Abstraction

Arrays-of-Structures or Structures-of-Arrays? What's the difference? Give an example.

Handwritten notes illustrating the difference between Arrays-of-Structures (AoS) and Structures-of-Arrays (SoA).

AoS (Arrays-of-Structures):

```
struct p { float x, y, z };  
pts[15][3];
```

SoA (Structures-of-Arrays):

```
pt.x[15] + pt.y[15] + pt.z[15]
```

The diagram shows three vertical bars representing arrays for x, y, and z. To the right, a horizontal bar represents a single structure containing x, y, and z. A bracket above the horizontal bar is labeled "AoS".

Language aspects of the distinction? Salient example?

complex

$x * y_i$

C and Multi-Dimensional Arrays: A Saving Grace

// YES:

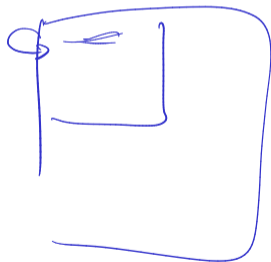
```
void f(int m, int n, double (*)[m][n]);
```

// NO:

```
struct ary {  
    int m;  
    int n;  
    double (*array)[m][n];  
};
```

// YES:

```
struct ary {  
    int m;  
    int n;  
    double a[];  
};
```



SIMD

Name language mechanisms for SIMD:

- intrinsics
- inline assembly
- `__m128`
- `#pragma`
- vector data types

[Demo: machabstr/Ways to SIMD](#)

Outer-Loop/inner-Loop Vectorization

Contrast *outer-loop* vs *inner-loop* vectorization.



Side q: Would you consider GPUs outer- or inner-loop-vectorizing?