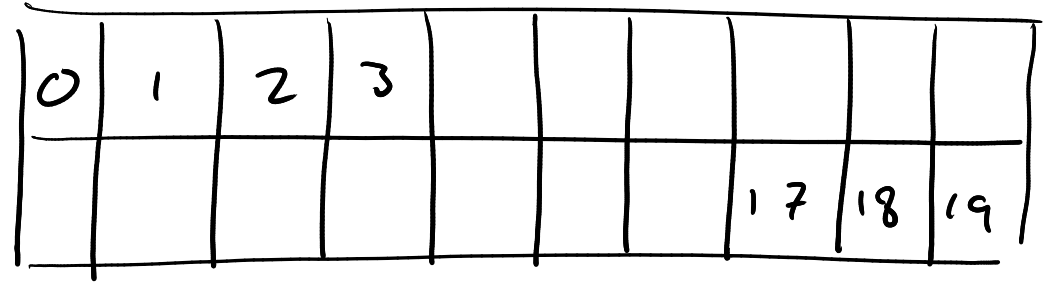**EE 200** Lecture 7:
Arrays of arrays and pointers

Steven Bell
28 September 2023

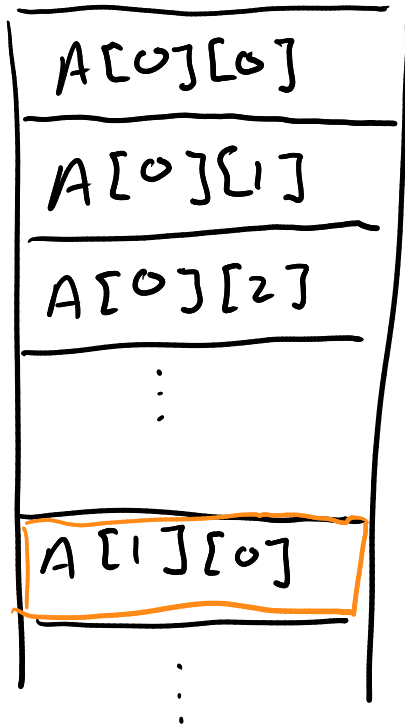Tufts
UNIVERSITY

Memory layout

Suppose we have the declaration

    int A[2][10]

Draw a diagram showing how this is laid out in memory

0	1	2	3				17	18	19

| A[0][0] |
| A[0][1] |
| A[0][2] |
| ⋮ |
| A[1][0] |
| ⋮ |

$$A + 1 \cdot (\text{row size}) + 0 \cdot (\text{element size})$$

$\underline{1} = \text{row \#}$ $\underline{0} = \text{col \#}$

Memory layout

Suppose we have the declaration

    int A[2][10]

What is the type of:

    A

    *A

    **A
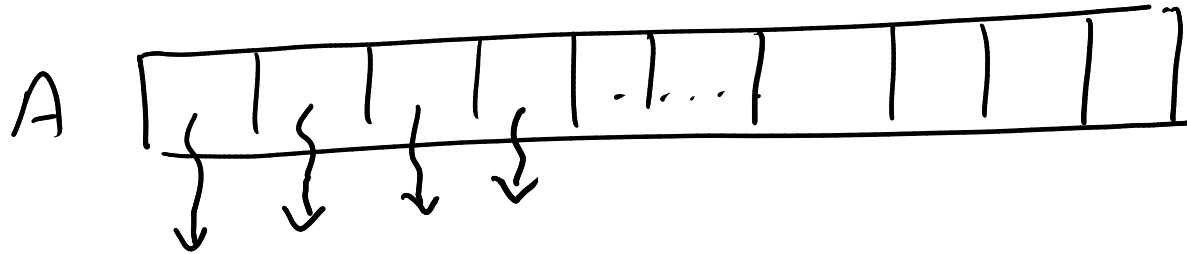
arraysizeof.c

Memory layout

Suppose we have the declaration

    int* A[10]
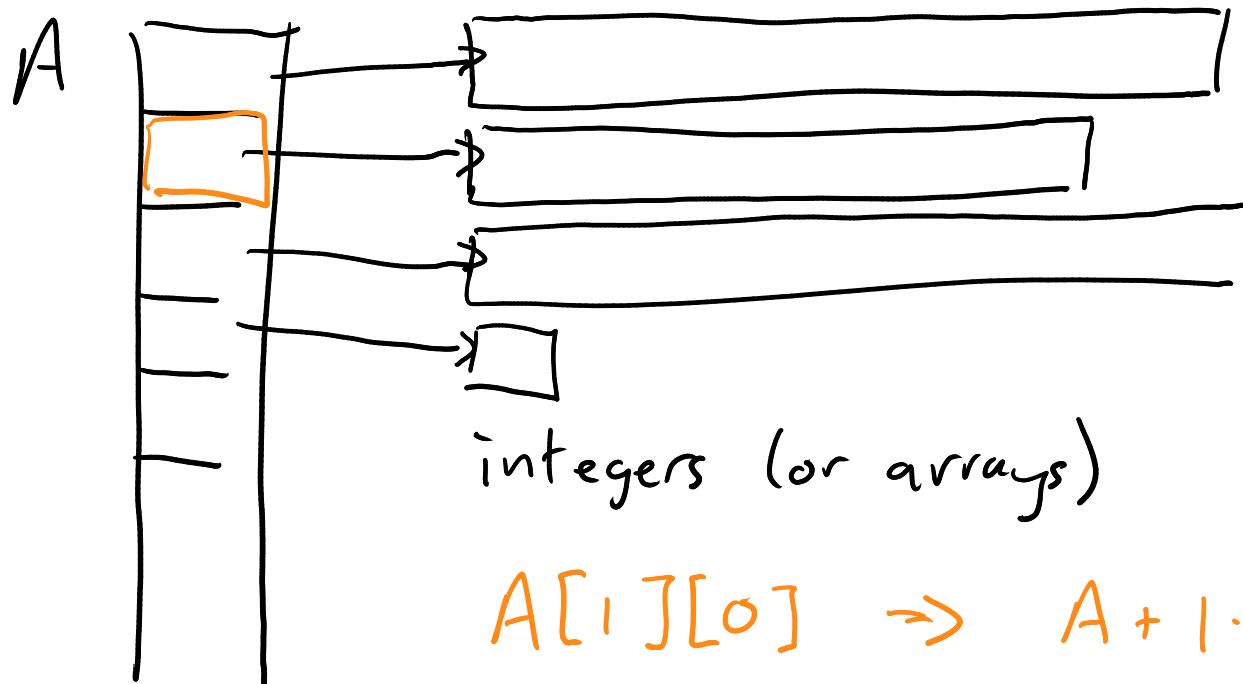
Draw a diagram showing how this is laid out in memory

Memory layout

Suppose we have the declaration

```
int* A[10]
```

Draw a diagram showing how this is laid out in memory

integers (or arrays)

+ Fine-grained memory management

– Can't iterate through

$A[1][0]$ => $A + 1 \cdot (\text{element size})$ dereference!
$0 \cdot (\text{element size})$

Pointers to functions?!

Nice if you want to pass a function to a function.

See SDL_AddTimer()

https://www.libsdl.org/release/SDL-1.2.15/docs/html/sdladdtimer.html

Make

Make provides a way to script your build process

```makefile
FLAGS=-lm
all: problem1
problem1: problem1.c test_problem1.c
    gcc problem1.c test_problem1.c -o problem1 $(FLAGS)
```

∧

Tab!

---

make *TARGET*   Build the named target

Upcoming deadlines

Homework 5 is due tomorrow at 11:59pm

Homework 4 resubmission due Tuesday at 4:30pm

Homework 6 posted, due Friday 10/6 at 11:59pm

Progress dashboard updated today (but not HW 2 or Perusall)