CSSE 220 Day 17

Data Structure Definition

Array implementation

Begin Data Structures Grand Tour

CSSE 220 Day 17

- Minesweeper team/team members peer review survey (on ANGEL) due by 5 PM Today.
- Home || Course > Lessons > Assignments > Minesweeper Evaluati...
- You will be asked to review several teams' Minesweeper programs for functionality issues before the end of the week.
  - More details later.
- Current Programming assignment: Hardy's Taxi. Due next Monday, but begin thinking about it yesterday!
  - An individual assignment.
- Markov assignment will be done in pairs. You can choose your partner again.
  - Must be different than your Minesweeper partner.

Hardy Grading Script ...

- ... appears to be ready. Let me know if you have any problems with it.

```
addiator 4:53am > cd /class/csse/csse220/200820/
addiator 4:55am > ./check Hardy
Checking Hardy
Clearing
/afs/rh/class/csse/csse220/200820/turnin/mrozekma/Hardy/extract/
Copying *.java... done

Compiling project...
No compile errors found
mrozekma - Summary for Hardy
Graded on Tue Jan 15 04:55:28 EST 2008

N    Points  Your Answer
1    15/15     1729 = 1^3 + 12^3 = 9^3 + 10^3
5    18/18     32832 = 4^3 + 32^3 = 18^3 + 30^3
30   10/10     515375 = 15^3 + 80^3 = 54^3 + 71^3
100  4/4     4673088 = 25^3 + 167^3 = 64^3 + 164^3
500  3/3     106243219 = 307^3 + 426^3 = 363^3 + 388^3

Points earned: 50/50
```

Answers to your questions

- Abstract Data Types
- Hardy's Taxi
- Material you have read
- Anything else

Today's agenda

- Binary Integer ADT exercise (with a partner)
- More big-oh practice
- Abstract Data types and Data Structures

For the next 35 minutes

- Work on the BinaryInteger exercise (linked from the Schedule page)
- Work with a partner (stand up …)
- If you finish early, work on Hardy's Taxi or the written homework problem from HW17

Practice: Apply the limit property to the following pairs of functions

1. N and N^2
2. $N^2 + 3N + 2$ and N^2
3. $N + \sin(N)$ and N
4. $\log N$ and N
5. $N \log N$ and N^2
6. N^a and N^N
7. **a^N and b^N $(a < b)$**
8. **$\log_a N$ and $\log_b N$ $(a < b)$**
9. $N!$ and N^N

Data and Abstract Data Types (Recap)

- What is data? (bits!)
- What is a Data Type
  - An interpretation of the bits
    - basically a set of operations

- Abstract Data Type example: non-negative integer
  - ZERO, succ, pred, isZero (derived methods plus, mult).
  - 1st representation: unary strings
    - ZERO is "", succ(zero) is "1", succ(succ(zero)) is "11"
    - We wrote succ( ) and pred( )
  - 2nd rep: **binary strings** (least-significant bit first)
    - ZERO is "0", succ(zero) is "1", succ(succ(zero)) is "01"
    - We wrote succ( )

Data Structures

- Most of the time when we talk about a **data structure**, we mean an ADT for storing several items (usually all of the items have the same type).
- When studying a new data structure, consider three aspects:
  - Specification (interface for the operations)
  - Implementation (sometimes several alternate implementations)
  - Application (how can it be used?)
- **Mostly, these can be considered independently.**
  - If we understand the interface and trust the person who says she implemented it, we can feel free to apply it without having to understand the details of the implementation.
- **220 emphasizes specification and application.**
- **230 emphasizes specification and implementation.**

The most common collection data structure is …

- An array.
- Size must be declared when the array is constructed
- We can look up or store items by index

```
a[i+1] = a[i] + 2;
```

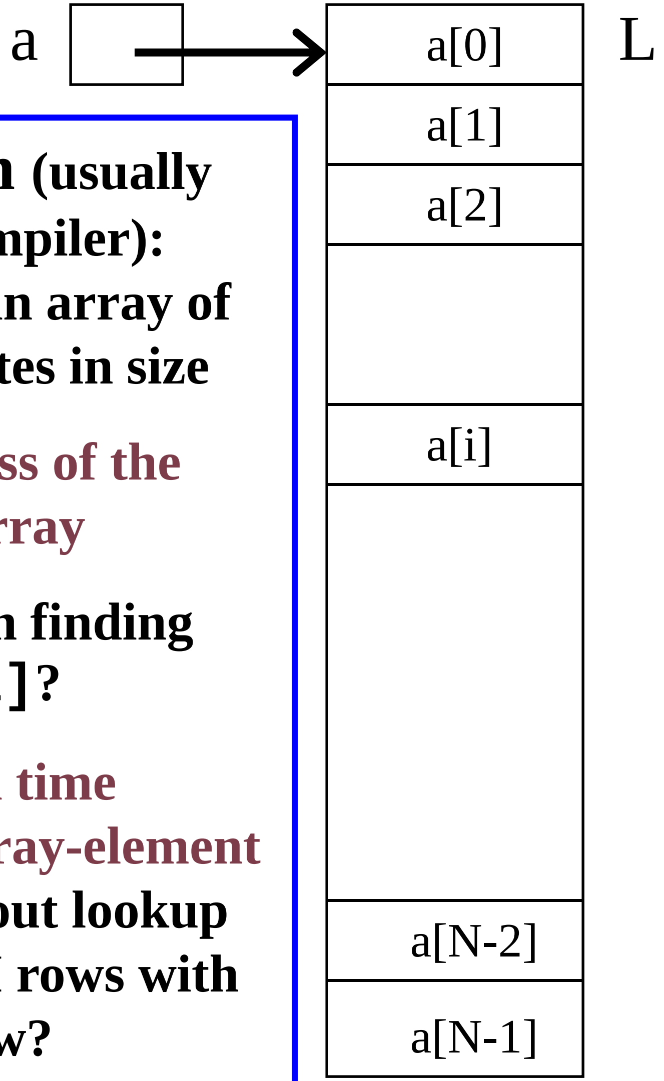
Implementation (usually handled by the compiler): Suppose we have an array of N items, each b bytes in size

Let L be the address of the beginning of the array

What is involved in finding the address of `a[i]`?

What is the Big-oh time required for an array-element lookup? What about lookup in a 2D array of M rows with N items in each row?

What about lookup in a 3D array (M x N x P)?



Some basic data structures

What is "special" about each data type?

What is each used for?

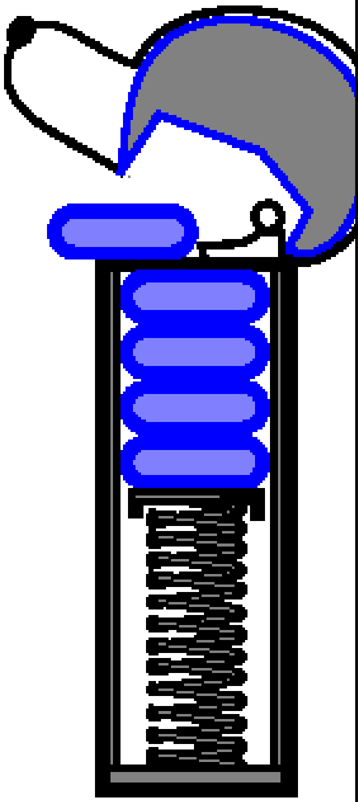
What can you say about time required for
 - adding an element?
 - removing an element?
 - finding an element?

- Array (1D, 2D, …)
- Stack

You should be able to answer all of these by the end of this course.

Stack

- Last-in-first-out (LIFO)
- Only top element is accessible
- Operations: push, pop, top, topAndPop
  - All constant-time.
- Easy to implement as a (growable) array with the last filled position in the array being the top of the stack.
- Applications:
  - Match parentheses and braces in an expression
  - Keep track of pending function calls with their arguments and local variables.
  - Depth-first search of a tree or graph.



Some basic data structures

What is "special" about each data type?

What is each used for?

What can you say about time required for
- adding an element?
- removing an element?
- finding an element?

- Array (1D, 2D, …)
- Stack
- Queue

You should be able to answer all of these by the end of this course.

Queue

- First-in-first-out (FIFO)
- Only oldest element in the queue is accessible
- Operations: enqueue, dequeue
  - All constant-time.
- Can mplement as a (growable) "circular" array
  - http://maven.smith.edu/~streinu/Teaching/Courses/112/Applets/Queue/myApplet.html
- Applications:
  - Simulations of real-world situations
  - Managing jobs for a printer
  - Managing processes in an operating system
  - Breadth-first search of a graph

Some basic data structures

What is "special" about each data type?

What is each used for?

What can you say about time required for
- adding an element?
- removing an element?
- finding an element?

- Array (1D, 2D, …)
- Stack
- Queue
- List
  - ArrayList
  - LinkedList
- Set
- MultiSet
- Map (a.k.a. table, dictionary)
  - HashMap
  - TreeMap
- PriorityQueue
- Tree
- Graph
- Network

You should be able to answer all of these by the end of this course.

Fixed-length Queue

- Specialized data structure.
- Useful for Markov problem.
- You and a partner should implement it in the next 25 minutes.
- Do it with another person.
- Put both people's names in a comment at the top of your program file.
- If you don't finish it now, finish it later today.
- Then we'll take a 5-minute break.

Some basic data structures

What is "special" about each data type?

What is each used for?

What can you say about time required for
- adding an element?
- removing an element?
- finding an element?

- Array (1D, 2D, …)
- Stack
- Queue
- List
  - ArrayList
  - LinkedList
- Set
- MultiSet
- Map (a.k.a. table, dictionary)
  - HashMap
  - TreeMap
- PriorityQueue
- Tree
- Graph
- Network

You should be able to answer all of these by the end of this course.