

ACSL Contest Topics

Dr. Baldassano

chrisb@princeton.edu

Yu's Elite Education

Next week

- ▶ Next week we are going to have two quizzes:
 - ▶ First half: ACSL Programming Contest
 - ▶ Second half: Final quiz for the class
- ▶ Today: review ACSL topics and the class

ACSL

- ▶ ACSL = “American Computer Science League”
- ▶ Runs programming and computer science competitions each year
- ▶ Yu’s is starting to compete next week - you will be the first competitors!

- ▶ Let’s review some things that will be on the ACSL contest:

What does this program do?

- ▶ You'll be given a program with a bunch of if statements
- ▶ You will have to keep track of all the variables and what gets printed out at the end
- ▶ Symbols to know:
 - ▶ Relational operators: `<`, `>`, `<=`, `>=`, `==`, `<>`
 - ▶ Logical operators: `and`, `or`
 - ▶ Math operators: `+`, `-`, `*`, `/`, `^`
 - ▶ Functions: `int()` and `print()`
 - ▶ `GOTO`

Program example

a = 4: b = 1: c = 3: d = 1: e = 0

if (a >= e) or (d < b) then a = e else b = d

if a >= b then a = a + b else a = a - b

if (b >= c) and (d <= e) then c = b - c else d = a - e

if b ^ d = 2 then d = d + 1 else b = b + 1

if int(a / c) = a / c then a = a / c else a = a - c

if a * e == c * d then a = a + d else c = c + e

print b + a * e / d - c * (a ^ a)

Binary numbers

- ▶ What are binary numbers?
- ▶ How can we convert to and from decimal numbers?

Binary numbers

The background of the slide is white with abstract green geometric shapes on the right and bottom edges. These shapes consist of overlapping triangles and polygons in various shades of green, from light to dark, creating a modern, layered effect.

Octal and hexadecimal

- ▶ What are octal and hexadecimal numbers?
- ▶ How can we convert between oct/hex and decimal?
- ▶ How can we convert between binary and oct/hex?

Octal and hexadecimal



Adding binary numbers



Multiplying by powers of 2



Number systems example

- ▶ Convert BED from hexadecimal to octal

Number systems example

Solve for X_2

$$X_2 = A12_{16} - 567_8$$

Functions

- ▶ What is a (math) function?
 - ▶ Takes a number as input, gives a number as output
 - ▶ Might do different things to different numbers

Recursive Functions

- ▶ Recursive function: Defined in terms of itself!

$$f(x) = \begin{cases} f(x-2) + 2 & \text{if } x > 2 \\ x + 2 & \text{otherwise} \end{cases}$$

Recursive example

- ▶ $f(x) = \begin{cases} f(x-20) + 10 & \text{if } x > 100 \\ 3x & \text{if } x \leq 100 \end{cases}$
- ▶ $f(150) =$

Recursive example

- ▶ $f(x) = \begin{cases} f(x+4)+2 & \text{if } x < 10 \\ x-8 & \text{else} \end{cases}$
- ▶ $f(f(5)) =$

Practice quiz

The slide features a white background with the text 'Practice quiz' in a green, sans-serif font. On the right side, there are several overlapping, semi-transparent green geometric shapes, including triangles and polygons, in various shades of green, creating a modern, abstract design.

Class review

- ▶ What is an algorithm? What is a data structure?
- ▶ What makes algorithms / data structures good or bad?

Class review

- ▶ Representing a set of numbers
 - ▶ Heaps:
 - ▶ What operations does it support? Big O?
 - ▶ How do we maintain a heap?

Class review

- ▶ Representing a set of numbers
 - ▶ Binary search tree
 - ▶ What operations does it support? Big O?
 - ▶ How do we maintain a binary search tree?

Class review

- ▶ Representing a set of numbers
 - ▶ Linked list
 - ▶ What operations does it support? Big O?
 - ▶ How do we maintain a linked list?

Class review

- ▶ Representing key->value pairs (associative array)
 - ▶ Hash table
 - ▶ What operations does it support? Big O?
 - ▶ What makes a good hash function?
 - ▶ What happens when hash function is good? Bad?

Class review

- ▶ Sorting
 - ▶ What are the most common sorting algorithms? Big O?
 - ▶ What about memory consumption and search stability?
 - ▶ Is it possible to beat the $O(N \log N)$ bound?

Class review

- ▶ Dynamic programming
 - ▶ What is the key property of a problem that allows for a dynamic programming solution?
 - ▶ What are some examples of dynamic programming?

Class review

- ▶ Machine learning
 - ▶ When do we use machine learning?
 - ▶ What is unsupervised machine learning?
 - ▶ What is supervised machine learning?
 - ▶ What are decision trees?
 - ▶ What is regression?

Class review

- ▶ Graph algorithms
 - ▶ What is a graph and why is it useful? Types?
 - ▶ Finding shortest path: Dijkstra's algorithm
 - ▶ Adding a heuristic: A*
 - ▶ Node importance: Pagerank
 - ▶ Strongly connected components: Tarjan
 - ▶ Minimum spanning tree: Prim's

Class review

- ▶ Game playing
 - ▶ Adversarial search
 - ▶ Minimax algorithm
 - ▶ How can we speed up minimax but still get exact solution?
 - ▶ How can we get an approximate solution?

Class review

- ▶ Procedural generation
 - ▶ What is the goal of procedural generation algorithms?
 - ▶ What is a Markov chain?
 - ▶ What is an L-system?