COMP20007 Design of Algorithms

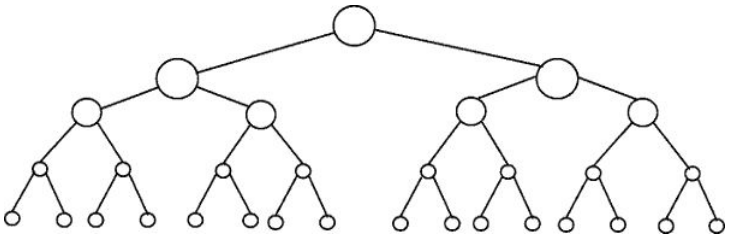
Week 2

Demonstrating Aikuma Smartphone App in the Brazilian Amazon

Administrivia

1. Tute and lab attendance
2. Lecture capture
   - sort of works but doesn't get the document camera
3. Assessment:
   - handbook says project will be due in week 4
     - we want to give you til end of week 5
     - two implementation methods
   - mid-semester test is worth 10%
     - use it to lock in marks
     - if you do better in the final exam, we'll scale that
     - max(MST + Exam, 70/60 * Exam)
4. Office hours:
   - Tuesday 11, Wednesday 2:15
   - Friday 10 (this week only)
   - other times by appointment

Theoretical limits: e.g. sorting



- "Guess the number I'm thinking of"
  - integers 1..2^p
  - how many questions do you need?

- Apply this idea to sorting n items: A, B, C, …
  - "guess how I've scrambled the sorted list"
  - decision tree for n=2:
    AB BA
  - decision tree for n=3:
    ABC ACB BAC
    BCA CAB CBA
  - Stir
    $$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$
  - We'll return to this in week 12

More examples of divide-and-conquer: Matrix multiplication

- Method 1

$$XY = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

- Method 2 (Strassen 1969)

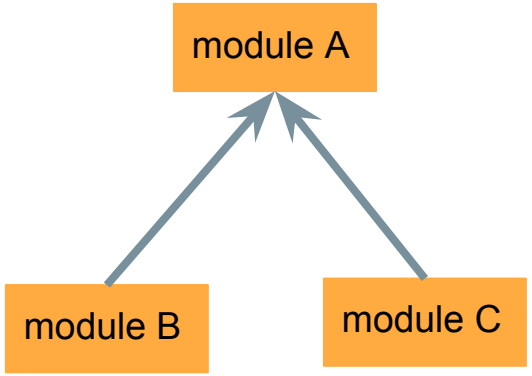
$$XY = \begin{bmatrix} P_5+P_4-P_2+P_6 & P_1+P_2 \\ P_3+P_4 & P_1+P_5-P_3-P_7 \end{bmatrix}$$

$$\begin{aligned} P_1 &= A(F-H) \\ P_2 &= (A+B)H \\ P_3 &= (C+D)E \\ P_4 &= D(G-E) \end{aligned}$$

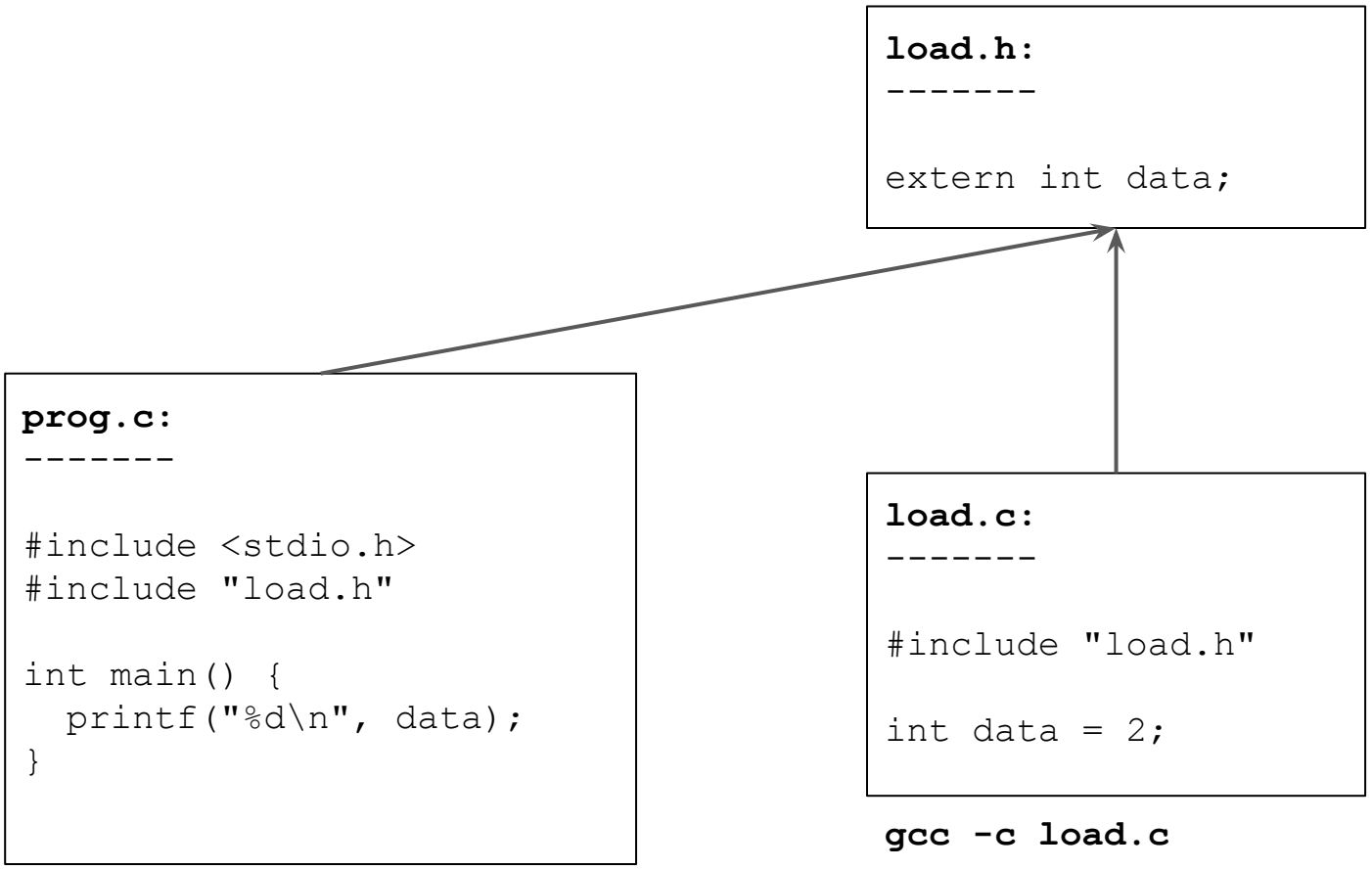
$$\begin{aligned} P_5 &= (A+D)(E+H) \\ P_6 &= (B-D)(G+H) \\ P_7 &= (A-C)(E+F) \end{aligned}$$

Divide and Conquer revisited: multi-module programs

- “let’s break it down”
  - benefits of modularity?
- architecture diagram
- how this is done in C
  - command line (bash, MinGW)
  - extern
  - Makefiles



A basic multi-module C program



gcc -o prog prog.o load.o (linking)

Makefiles, dependencies, and toposort