Administrativia

Mark Allman
mark.allman@case.edu

EECS 325/425
Nov 12 2018

"The sun is hot and that ol' clock is movin' slow …
… and so am I …"

Docket

	This Week	Next Week	Next Week++
Mon	**Guest Speaker (Joseph Ishac, NASA GRC)**	Lecture: CC, CDN, …?	Lecture: ?
Tue			
Wed	Lecture: DNS Project #4 Due Project #5 Assigned	Lecture: CC, CDN, …? ?? Return Proj #4 ??	Lecture: ? Project #1 Due
Thu		Thanksgiving	
Fri			

Project 4 Reminder

- Ugrads: remember the extra credit
  - doesn't have to be overly involved
  - not available on projects submitted late

Project 4: -m

Project 4: -m

- Path #1: use ‘map’ data structure from the C++ STL

Project 4: -m

- Path #1: use ‘map’ data structure from the C++ STL

- Path #2: read chapter 6.6 of K&R (“table lookup”)
  - goes through implementation of a hash table

Project 4: -m

Project 4: -m

- Deadlines force sub-optimality …

Project 4: -m

- Deadlines force sub-optimality …

- *IF* you're options are …

Project 4: -m

- Deadlines force sub-optimality …

- *IF* you're options are …

1. work on appropriate data structure and possibly lose correctness and stress points

Project 4: -m

- Deadlines force sub-optimality …

- *IF* you're options are …

1. work on appropriate data structure and possibly lose correctness and stress points

2. use a cheap and suboptimal data structure to get correctness points, but for sure sacrifice the stress points

Project 4: -m

- Deadlines force sub-optimality …

- *IF* you're options are …

  1. work on appropriate data structure and possibly lose correctness and stress points

  2. use a cheap and suboptimal data structure to get correctness points, but for sure sacrifice the stress points

- … you might choose the latter (at some point)

Project 4: -m

Project 4: -m

- Old adage: Make it work. Then make it work fast.

Project 4: -m

- Old adage: Make it work. Then make it work fast.

- Begrudging Path #3:
  - use an array of structs (src, dst, counter)

Project 4: -m

- Old adage: Make it work. Then make it work fast.

- Begrudging Path #3:
  - use an array of structs (src, dst, counter)
  - exploit locality

Project 4: -m

- Old adage: Make it work. Then make it work fast.

- Begrudging Path #3:
  - use an array of structs (src, dst, counter)
  - exploit locality
    - remembering the last two places you inserted will handle 41% of the lookups neede for insertion (desperado.trace)

Project 4

- -s: summary mode
- -l: length mode
- -p: packet printing mode
- -m: traffic matrix mode

Project 4

- ✓ -s: summary mode
- ✓ -l: length mode
- ✓ -p: packet printing mode
- ✓ -m: traffic matrix mode

Questions ??

Joseph Ishac

- Computer engineer @ NASA GRC
- Case alum (BS & MS)