

1  CS3134 #25

12/4/03

Janak J Parekh

2  Administrivia

- Last lecture!
- Reminder: course evaluations
- HW6 due on Monday
  - Minor clarifications; redownload
- HW5 return next week
- Extra exam
  - Thursday, 11am-2pm
- Final review session
  - Tuesday, 2pm-4pm

3  Agenda

- Do Floyd's once again
  - My algorithm last time on the board was (slightly) wrong
- Discuss HW#6
- Tie things together
- What will the final exam be?

4  Floyd's Algorithm

- For all-pairs shortest path, in  $V^3$  time
- Idea based on Warshall's algorithm, but *add* weights together
- For all rows  $y$ ,
  - For all columns  $x$  in row  $y$ ,
    - If any value  $(x,y)$  is 1,
    - For all rows  $z$  in column  $y$ ,
      - If  $(y,z) + (x,y)$  is less than  $(x,z)$ , then update  $(x,z)$
      - Optionally, store path  $(x,z)$  through  $y$
- That's it!
  - Remember array references are "backwards"  $[y][x]$

5  Putting it all together...

- What have we studied?
- Low-level structures
  - Arrays, references
- High-level structures
  - Lists, hash tables, trees, graphs
- Algorithms
  - Recursion
  - Insertion sort, Quicksort, Mergesort, Heapsort
- Multiple ways to slice-and-dice
  - Book: "general-purpose" vs. "specialized"
- Nifty tables on pgs 722, 724, 725

6  Intractable problems

- There are graph (and other!) problems that can't be done in any reasonable time (linear, logarithmic, polynomial) – they're often exponential time, e.g.,  $x^n$  – and grow way too quickly
- Considered NP-complete (Non-deterministic Polynomial)
- Insta-Ph.D.: prove  $P=NP$  (or vice-versa)
- Example: traveling salesman problem -- visit all cities exactly once, and return to starting point, taking minimum-cost path
  - Hamiltonian cycle problem
  - $N!$  time!

## 7 Java data structures

- Collections (container) API
- Collections and maps
  - Collections: Sets, SortedSets and Lists
  - Maps: Map and SortedMap
- Implementations:
  - Sets: HashSet, TreeSet
  - Lists: ArrayList, LinkedList
  - Maps: HashMap, TreeMap
- Lots of utility methods
  - Sort, shuffle, search, findMax/findMin
- Works with generic "Object"s
- In the real world, get comfortable with these – they work well!

## 8 The Exam

- Similar to midterm, but about 50-75% longer
- What you don't need to know
  - Shellsort
  - Red-black trees
  - 2-3-4 trees/external storage
  - Floyd's algorithm (too hard to do on the exam)
- What you do need to know
  - Pretty much everything else
  - Remember, stuff in class – use my slides
- Chapter 15 is a useful overview

## 9 Next time

- Review session
- Final