

CSSE 220

Merge Sort

Checkout *MergeSortSimple* project from SVN

Today's Plan

- Big-oh practice
- Merge sort

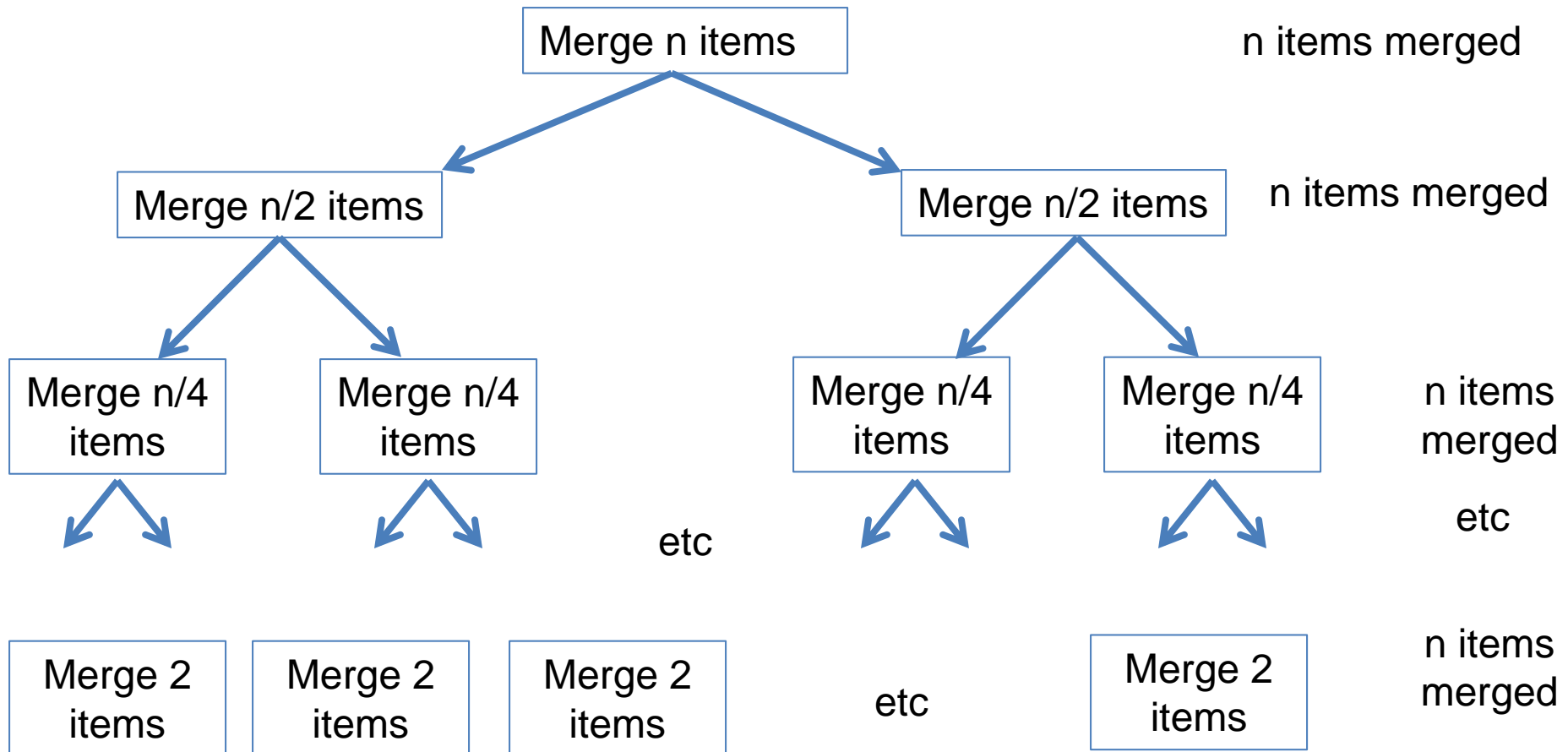
Merge Sort

- Basic recursive idea:
 - If list is length 0 or 1, then it's already sorted
 - Otherwise:
 - Divide list into two halves
 - Recursively sort the two halves
 - **Merge** the sorted halves back together

Analyzing Merge Sort

If list is length 0 or 1,
then it's already sorted

- Otherwise:
 - Divide list into two halves
 - Recursively sort the two halves
 - **Merge** the sorted halves back together



Next Week: Your Presentations

- 8 minutes long
- ~3 minutes showing off your extra features
- ~5 minutes explaining JUST ONE technical decision in detail
 - Could be a design decision, feature implementation, or tricky bug
 - Be careful about code examples (good – but in moderation)
 - Should include prepared slides usually including diagrams

Arcade Game Presentation

So the arcade game presentation is a way for you to show off your cool work and also show us (the professors) that you understand some of the hard issues we've been trying to teach you. In the presentation it's most important you do two things:

1. Showcase your special features. Remember that everyone in your class has built the same game, so there's no need to go through all the regular stuff in detail. Time: approximately 1/3 of your talk.
2. Talk about 1 interesting design challenge you had, and how you solved it. Time: approximately 2/3 of your talk.

Here's how the grading works (25 points total):

5 points: All team members in attendance. Each team member should also participate in the presentation in some way.

5 points: Team dressed up. You must dress nicely for the talk. Each team member should be in "business casual attire". Collared shirts, no jeans, etc. As long as I can tell you tried, you're probably OK.

5 points: Polished presentation, right length. Usually these talks are 10 minutes/team but depending on the class size it may be slightly different. It should be clear that you prepared for this talk and practiced. People should know their roles and the presentation should take the full time but not go over.

5 points: Clear design presentation. Explaining a design issue is difficult. Looking at a giant UML diagram with all your classes is usually not helpful, nor is just quickly moving through a whole bunch of code real quickly. What you'll want to explain is a UML diagram that highlights ONLY the issue you want to present, or slides with code that highlight the key section. One way or another, it should be clear that you've given thought to how to present your design issue in a good way.

5 points: Correct/Interesting design issue presented. You need to choose a topic that has a genuinely interesting design problem, discuss the tradeoffs well and correctly, and then come to a sane conclusion.