Mazes and Superior Sorts

11-29-2010

Opening Discussion

- Solutions to the IcP.
- Minute essay comments
  - Translating human thoughts into computer logic.
  - Learning to write recursive functions.
  - How do we make recursive functions call themselves more than once?
- Watching the Hanoi algorithm.

Mazes

- My favorite example of recursion is mazes.
- Consider a maze as a 2-D grid with each square either filled or not.
- Now the challenge is to find the length of the shortest path through the maze.
- How do you do that?

Superior Sorts

- We can also use recursion to write some better sorts.
- All of our old sorts could have been written with recursion, but only as a substitute for iteration.
- With recursion we can do sorts that work by repeatedly breaking the set down then work recursively on the pieces.
- Do they do the work on the way down the stack or back up?
- Work fairly well on lists.

Merge Sort

- Simple description
  - Break the collection in two and make a recursive call on the two halves.
  - Merge together the sorted results with an O(n) merge.
- Can't be done in place, but that is advantageous for lists which are immutable.
- O(n log n) all the time.

Quick Sort

- Description
  - Pick a pivot and move everything less than the pivot below and everything greater above.
  - Recurse on the two sides of the pivot.
- Can be done in place, but Scala collection methods allow very simple form that isn't in place. We'll wrote both.
- Speed depends on pivot selection. O(n log n) on average with random data, but can be as bad as $O(n^2)$ with bad pivots.

Minute Essay

- What problems could we have with our maze algorithm?
- Not IcP because of assignment.