

CSC207.01 2014S, Class 29: Implementing Lists with Arrays

Overview

- Preliminaries.
 - Upcoming work.
 - Admin.
- Some notes on Quicksort.
- Some notes on implementing lists with arrays.
- Lab.

Preliminaries

Get ready!

- Fork and clone the repository at <https://github.com/Grinnell-CSC207/lab-array-based-lists>

Upcoming Work

- Reading for Tuesday: Linked lists (forthcoming).
- Today's lab writeup: Exercise 6 (implement previous)
- Exam 2 to be distributed on Wednesday.

Admin

Some notes on Quicksort

Big Picture

- Use partition to divide the array into small and large stuff
- Recurse on each half
- Uses randomness and divide and conquer

Partition

- Randomly choose a pivot. Compare each element with that one. Anything less than or equal to the pivot goes to the left. Anything greater than the pivot goes to the right
- Returns the location of the pivot after partitioning

More on Partition

- How might you find the median of an array, using partition?
 - Knowledge: How many elements are in the array
 - Easy technique: Sort, use element `array.length/2`. Expect $O(n \log n)$
 - Harder technique: Partition once
 - If the pivot is at the midpoint, done
 - If the pivot falls after the midpoint, recurse on the left
 - If the pivot falls before the midpoint, recurse on the right

```
public T median(T[] values, Comparator order) { int midpoint = values.length / 2; int lb = 0; int ub = values.length; while (true) { int pivloc = partition(values, lb, ub, order); if (pivloc == midpoint) return values[midpoint]; else if (pivloc < midpoint) ub = pivloc; else lb = pivloc+1; } } // median
```

What's the running time? $n + n/2 + n/4 + n/8 + \dots \sim 2n$ $O(n)$

Is partition like any other algorithm?

- It's a simplified version of the Dutch National Flag problem.
- We can write Quicksort using DNF to partition rather than partition.

Some notes on implementing lists with arrays

- Key operations: Insert, Remove, Replace, Iterate
 - Also need a notion of "location"
- Implementing with arrays
 - Position: Index into the array
 - Iterate: Get the value at that index
 - Insert: Shove stuff to the right
 - Remove: Shove stuff to the left
 - Replace: Replace
- Fun with Java: Use ListIterators

Lab

Copyright (c) 2013-14 Samuel A. Rebelsky.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.