Announcements

- Program due Sat.
- Next HW out on Friday, due in 1 week
- Next Friday will be review session, exam the following Mon.
- Lab tomorrow
Q: Describe insertion, selection, or merge sort.

Q2: Name 2 other sorting algorithms.
Sorting Algorithms

Why do we care?

✓ - Insertion
✓ - Selection
✓ - Merge
✓ - Bubble
✓ - Quick
✓ - Bucket
✓ - Radix
✓ - Shell
- van Em de Boas
Smart Selection sort

Assume A[1..i] are sorted

take A[i+1]

Find where it goes ≤ binary search O(log i)

Put it there O(i) in vectors
Merge Sort \[ \log_2 n \]

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
9 & 10 & 11 & 12 \\
13 & 14 & 15 & 16 \\
17 & 18 & 19 & 20
\end{array}
\]

MergeSort( \[ 2 \quad 8 \quad 11 \quad 12 \] )

\[
-1 \quad 2
\]

\[ T(n) = O(n) + 2T(\frac{n}{2}) = O(n \log_2 n) \]
Bucket

$A[0] \ldots A[n-1] \leftarrow O(n)$

$n$ things in list

between 0 and $N$

$O(N)$

$\Rightarrow O(n+N)$
Application

Students: Alice
Bob
Zeb
Bill
Adam

Alice  Adam  Bob  Bill  Zeb

N = 26  Zeb
Radix Sort

Adam → Alice → Bill → Bob → Zeb
Practicalities

Experimentally, quicksort runs faster than merge on small inputs. Why?

Quicksort can be done "in place"
Merge sort has more parameter
Programmer time
Trees

Only inherently linear.

root → Abraham

leaves

2-dimensional
Node in a tree

- Object (data)
- Store children
- Parent

Tree class: _root

Binary tree:
- Left
- Right
Heaps
Binary Search Trees