

CS180 - Sorting + trees

Note Title

11/2/2011

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
 - ⋮
- ~~n^2~~ ~~$n \log n$~~ n^2 or $n \log n$

Smart Selection sort

Assume $A[1..i]$ are sorted

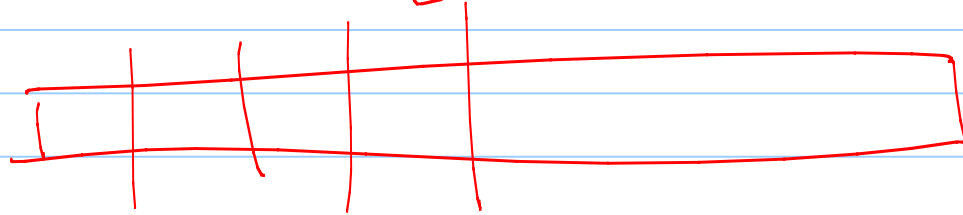
take $A[i+1]$

Find where it goes \leftarrow binary search
 $O(\log i)$

Put it there $O(i)$ in vectors

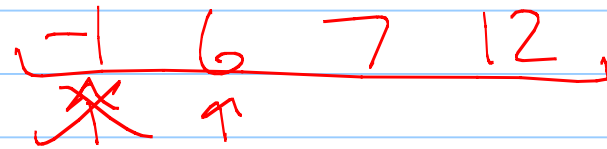
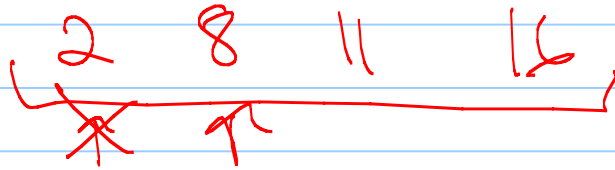
Merge Sort $\lfloor n/2 \rfloor$

1 2 3 4 11 12 16 18



$\lfloor 11 \rfloor$ $\lfloor 12 \rfloor$

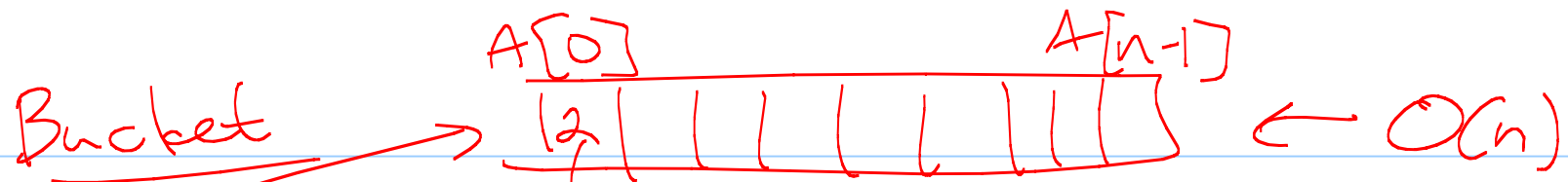
Mergesort (



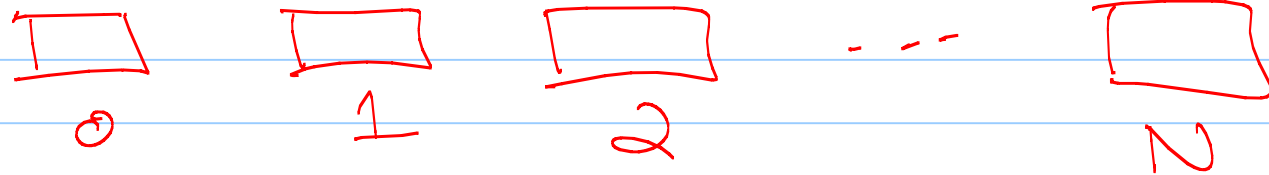
1 2

$$T(n) = O(n) + 2T\left(\frac{n}{2}\right)$$

$$= O(n \log_2 n)$$



n things in list
between 0 and N

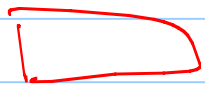


$O(N)$

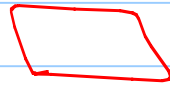
$\Rightarrow O(n+N)$

Application

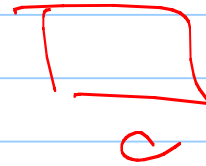
Students: Alice
Bob
Zeb
Bill
Adam



A ↓
Alice
↓
Adam



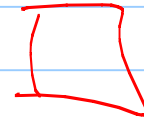
B ↓
Bob
↓
Bill



C

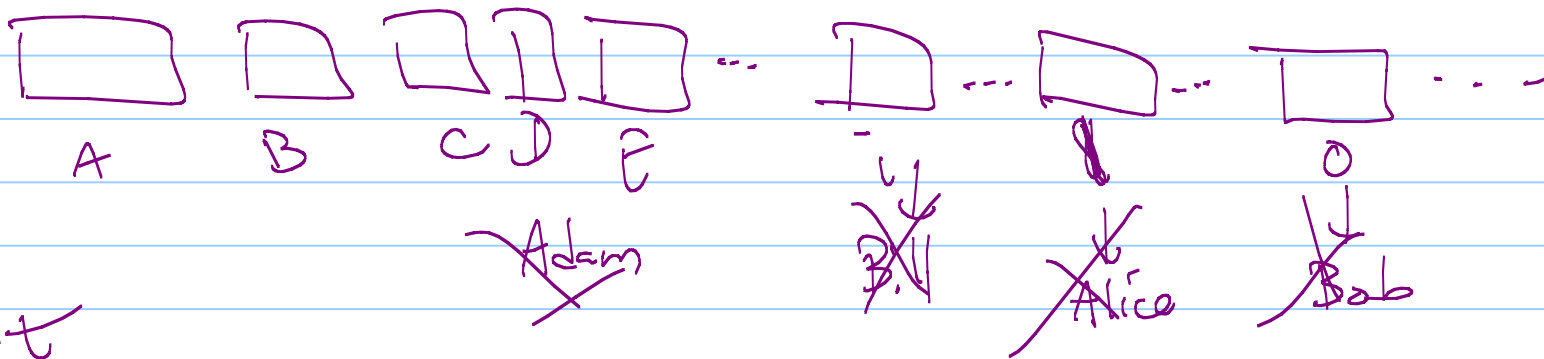
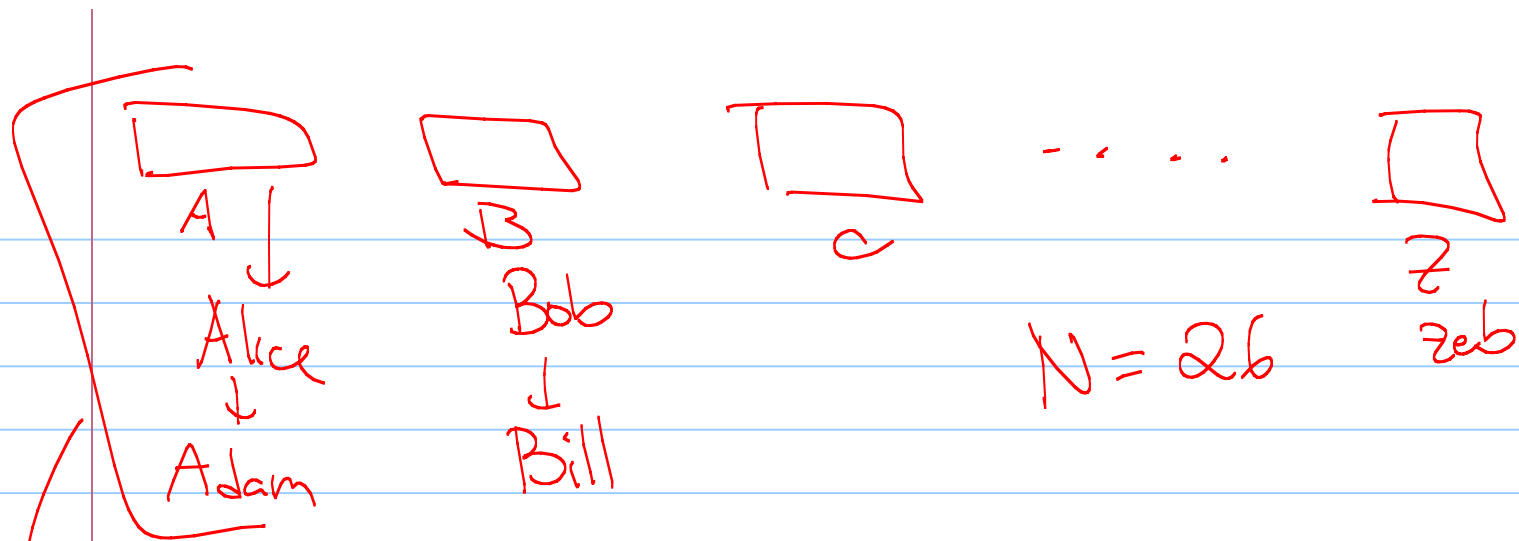
...

N = 26



Z
Zeb

Alice Adam Bob Bill Zeb



Radix Sort

Adam \rightarrow Alice \rightarrow Bill \rightarrow Bob \rightarrow Zeb

Practicalities

Experimentally, quicksort runs faster than merge on small inputs.

Why?

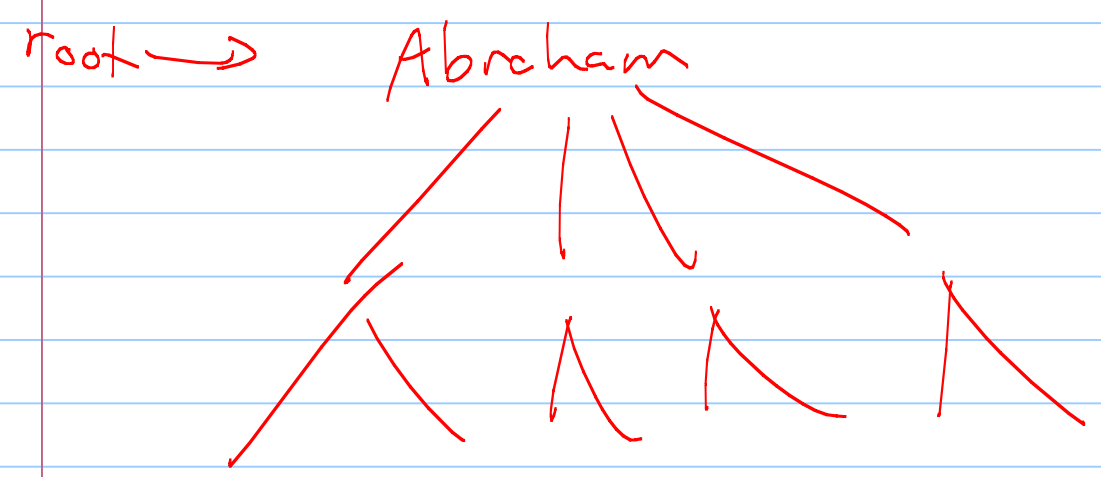
Quicksort can be done "in place"

Merge sort has more parameter

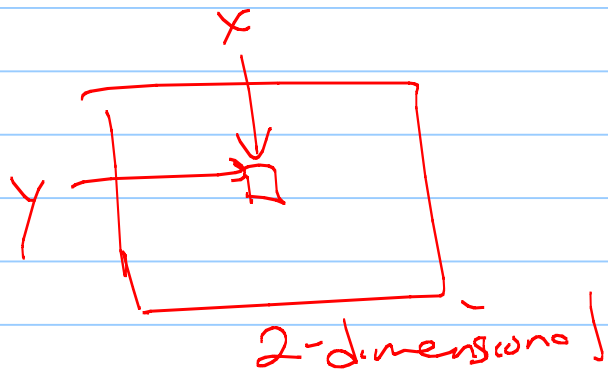
programmer time

Trees

Only inherently linear.



leaves →

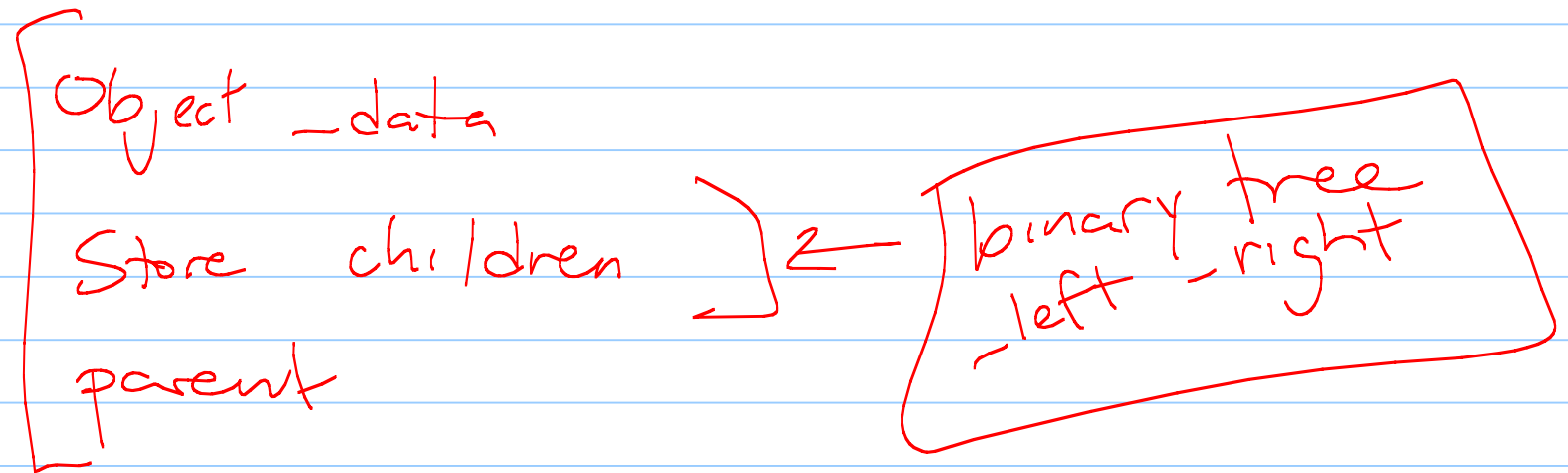
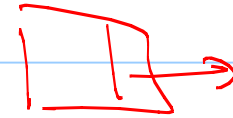


recharba5 ←

index.txt: Sk



Node in a tree



Tree class : _root

Heaps

Binary Search Trees