More on comparison-based sorting

Note Title 11/7/2007

Is it possible to sort in linear time?

What is the worst-case lower bound for comparison-based sorting?

$\Omega(n \log n)$

Do any comparison-based sorting algorithms have a linear best case?

Insertion Sort is $\Omega(n)$

Some features of comparison-based sorting:

- insertion sort does very well when input almost sorted
  - also practically good when n is small
- quicksort often fast in practice but it depends on the input and the pivots chosen

In addition to comparisons,

we also care about

- memory consumed
- # of reads
- # of writes

Heapsort and quicksort can be executed in place.

Can hybrid algorithms get us improved performance?

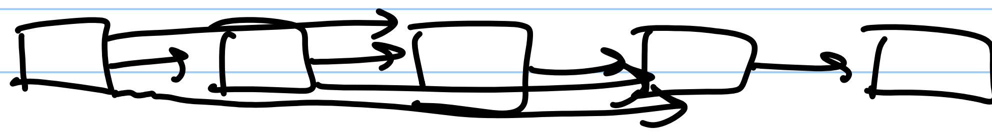
Example: Insertion Sort

- in iteration i, insert ith element into sorted sublist
- use binary search to find the place to put ith value in list

$O(\log n)$ to insert (comparisons) → $O(n)$ shifting

$O(n)$ insertions

$O(n \log n)$ in worst case??



Complications

1. duplicate elements
2. non-primitive keys

   First Name
   Last Name
   Test Score

   } what if you sort by LastName, First Name?

3. Data Stability

   Say you have list

(Ada Smith 75, Bob Jones 80, Clara Barton 70,
George Mason 90, Shia LeBouf 65,
Clara Barton 95)

Notice "Clara Barton" is in the list twice.
DATA STABILITY means that items with the same keys stay in the same relative positions.

input: (Ada Smith 75, Bob Jones 80, Clara Barton 70, George Mason 90, Shia LeBouf 65, Clara Barton 95)

STABLE OUTPUT

Clara Barton 70
Clara Barton 95
Bob Jones 80
Shia LeBouf 65
George Mason 90
Ada Smith 75

not stable output

Clara Barton 95 ←
Clara Barton 70 ← not in order input arrived in
Bob Jones 80
Shia LeBouf 65
George Mason 90
Ada Smith 75

Why is data stability useful?

Lower Bound of Comparison-Based Sorting is $n \log n$

- means <u>worst-case</u> of <u>any</u> algorithm that could ever be invented is $\Omega(n \log n)$.

Insertion Sort is $O(n^2)$
$\Omega(n)$.

↓

When you use $\Omega(f(n))$ to describe an algorithm,
you are saying on <u>all</u> inputs running time is <u>at least</u> $f(n)$.

→ best case is at least $f(n)$.

worst-case of Insertion Sort is $\Omega(n \log n)$, in fact it is $\Omega(n^2)$.