CS180 - Lists (recap) + Sorting

Note Title 10/31/2011

Announcements

- No class tomorrow (lab still due Sunday)
- Check point tomorrow
- HW due Sunday

Operator < (in vectors)

$[a_1\ a_2 \cdots a_n]$ $[b_1 \cdots b_m]$

compare a_1 & b_1

$A[i] > B[i]$

Vectors versus lists

Q: What would operator [] look like in a list?

start at _sent.

loop up to input # ⟶ O(n)

move to next

Vectors versus lists (cont)

Running times:

	Vectors	Lists
→ operator []	O(1)	O(n)
find	O(n)	O(n) ←
insert	O(n)	O(1)
erase/remove	O(n)	O(1)

Searching

What is linear search? $O(n)$

Hunt element by element.

Binary search?

Fundamental divide & conquer.

$B(n)$ = running time on input of size n (in vector)

$B(n) = 1 + B\left(\frac{n}{2}\right)$

$= O(\log_2 n)$

in list: $B(n) = \frac{n}{2} + B\left(\frac{n}{2}\right)$

Sorting

Name some sorting algorithms.

- ✓ - Bubble sort ←
- ✓ - Merge sort
- - Insertion sort
- ✓ - Quick sort
- ✗ - Marriage sort
- - Bucket sort
- ⋮

Bubble sort

Move down list comparing neighbors
If in wrong order, swap them

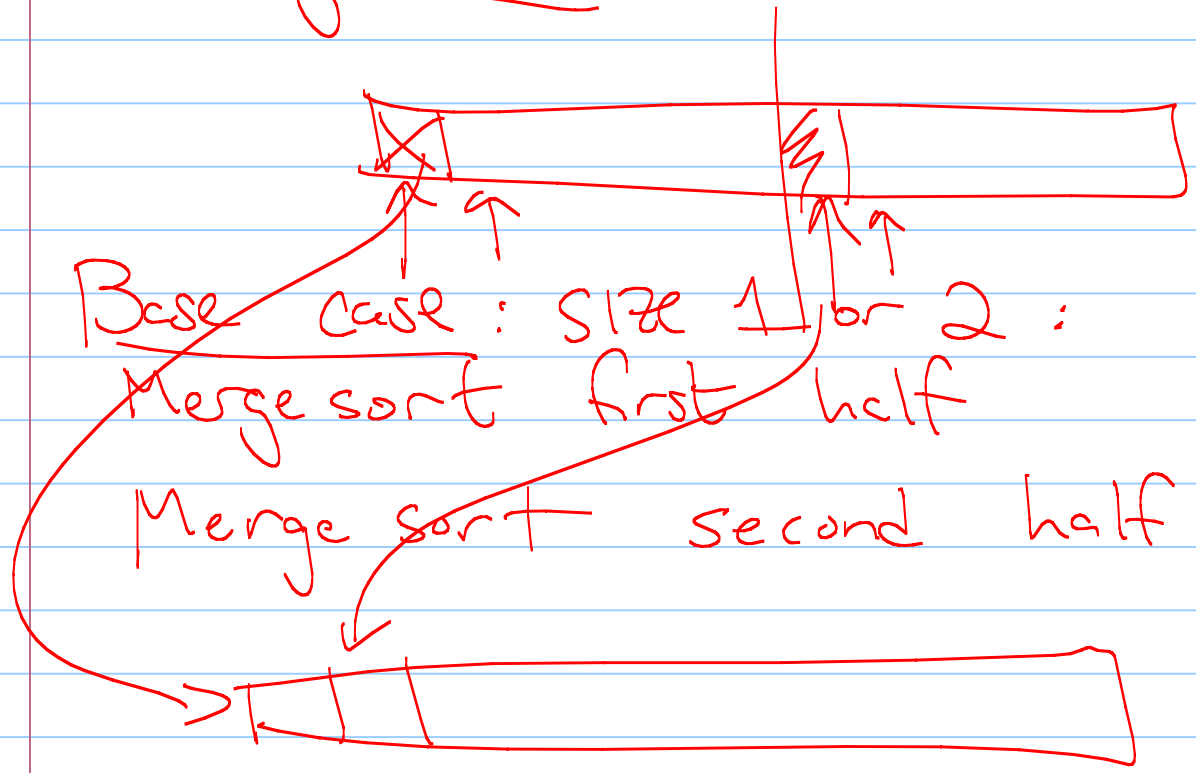
At end of list, largest element must be at end

(nested for loops)

Easy to code.

$O(n^2)$

Merge sort

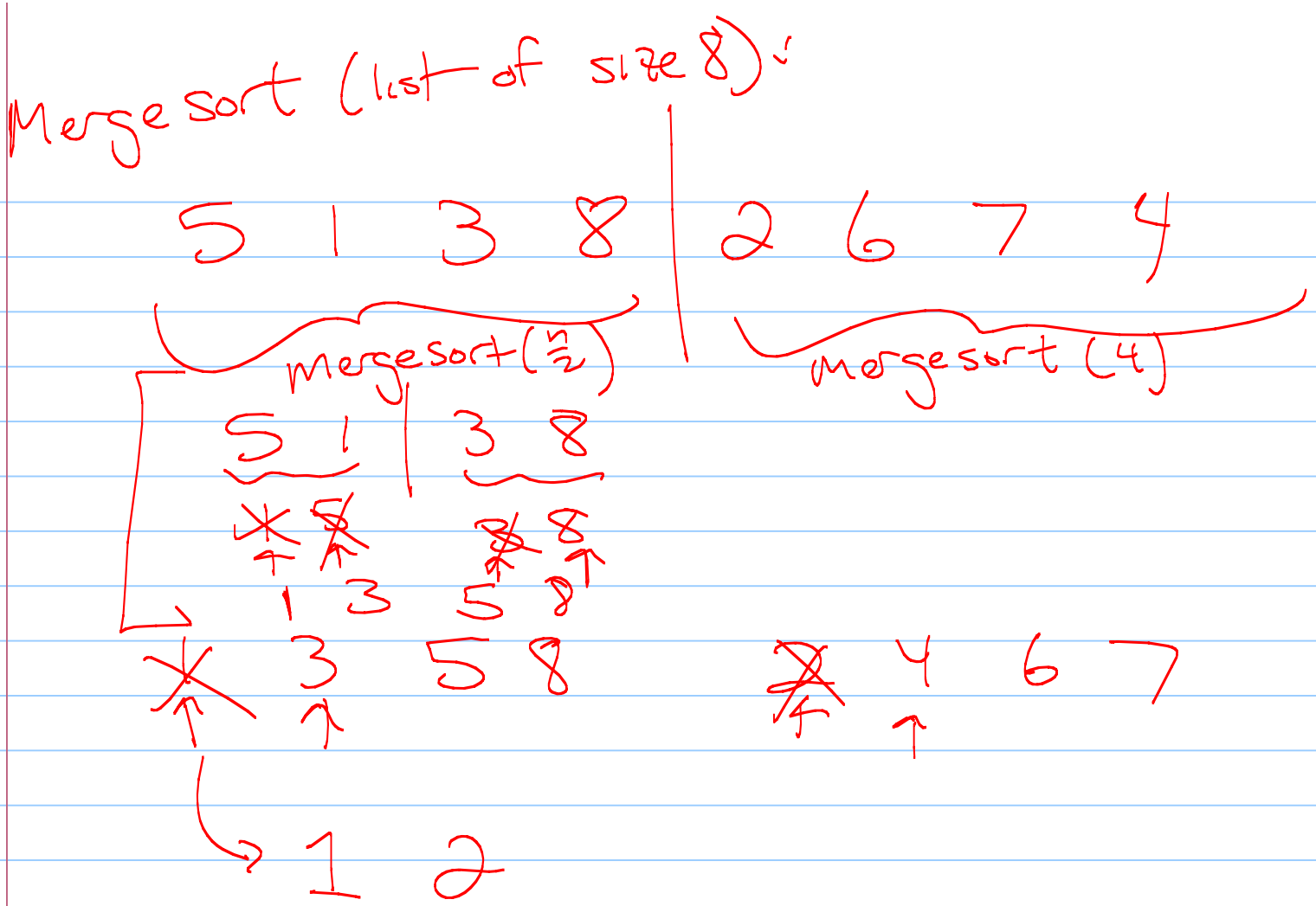


Base case: size 1 or 2:

Merge sort first half

Merge sort second half

Claim: At most n comparisons until sorted.


Mergesort (list of size 8):
5 1 3 8 | 2 6 7 4
mergesort(n/2)
mergesort(4)
5 1 | 3 8
1 3 5 8
1 3 5 8
4 6 7
1 2

Quicksort

10 2 3 11 6 9 -1 7 12

2 3 6 9 -1 7 10 11 12

Call Quicksort on this half

Quicksort

In worst case, get bad pivot.

Merge sort: $M(n) = M(\frac{n}{2}) + M(\frac{n}{2}) + O(n)$

$= 2M(\frac{n}{2}) + O(n)$

$= O(n \log n)$

Quicksort: $Q(n) \leq n + Q(n-1)$

$= \sum_{i=1}^{n} i = O(n^2)$

$n-1$

Insertion:

$k-1$ are in sorted order | k^{th}

for $k = 2$ to n

figure out where k belongs & put it there — linear

$$\sum_{k=2}^{n} \log k$$

$$= \log 2 + \log 3 + \cdots + \log n$$

$$= \log (2 \cdot 3 \cdot 4 \cdots n)$$

$$= \log (n!)$$

$$= O(n \log n)$$

$$O(n^2)$$