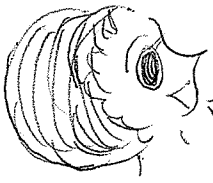


CS311 Analysis Summary

MARY



I search sequentially from the beginning to the end

$$W(N) = O(N) \\ B(N) = O(1) \quad \text{any list}$$

JOHN



I'd divide the list in $1/2$ to search

$$W(N) = O(\log_2 N) \\ B(N) = O(1) \quad \text{sorted list}$$

DR. X



you can't do better than this

$$F(N) = O(\log_2 N) \\ \text{search a sorted list}$$



DR. X

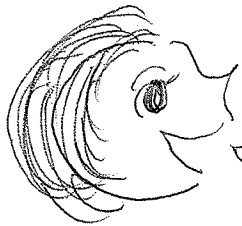
Want to find the largest? you can't do better than this.

$$F(N) = O(N)$$

$N-1$ must lose

Sorting

MARY



I used
the
Selection
Sort !!

$$W(N) = O(N^2)$$
$$B(N) = O(N^2)$$



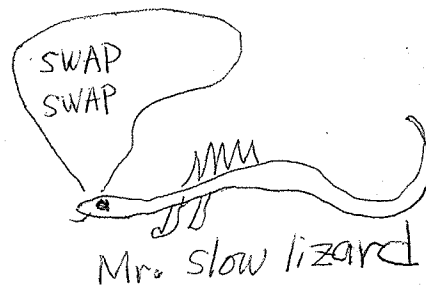
Mr. Slow Snake



JOHN

I used
the
Bubble
Sort !!

$$W(N) = O(N^2)$$
$$B(N) = O(N)$$



Mr. Slow Lizard



KATHY

I used
the
Insertion
Sort !!

$$W(N) = O(N^2)$$
$$B(N) = O(N)$$



Mr. skinny lizard



DR. X

You can't
do better
than this if
one correction
per
comparison.

$$F(N) = O(N^2)$$

You are
too slow for
a large list. Sorry

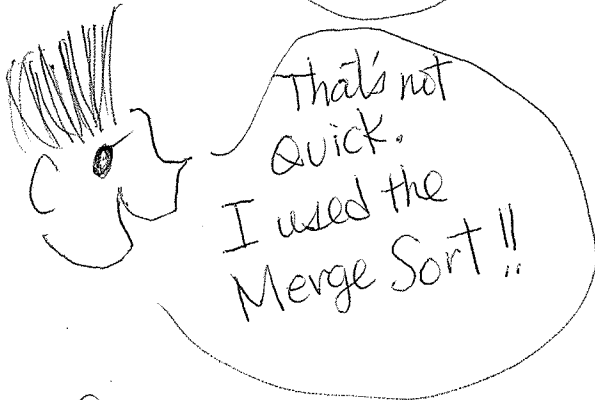
Divide & Conquer Sorting



$$W(N) = O(N^2)$$

$$B(N) = O(N \log_2 N)$$

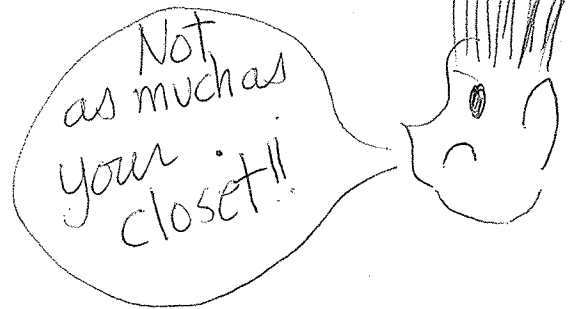
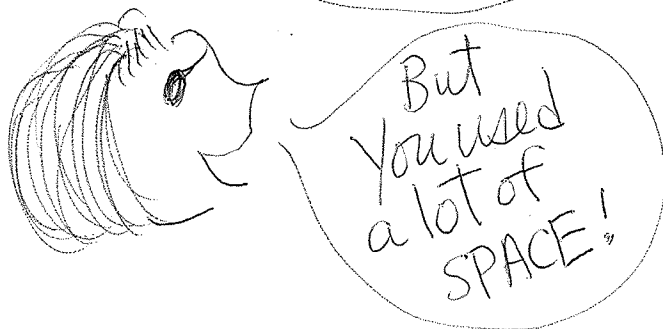
Partition
 $\frac{1}{2}$ each time
VS.
 $1 \approx N-1$ each time



$$W(N) = O(N \log_2 N)$$

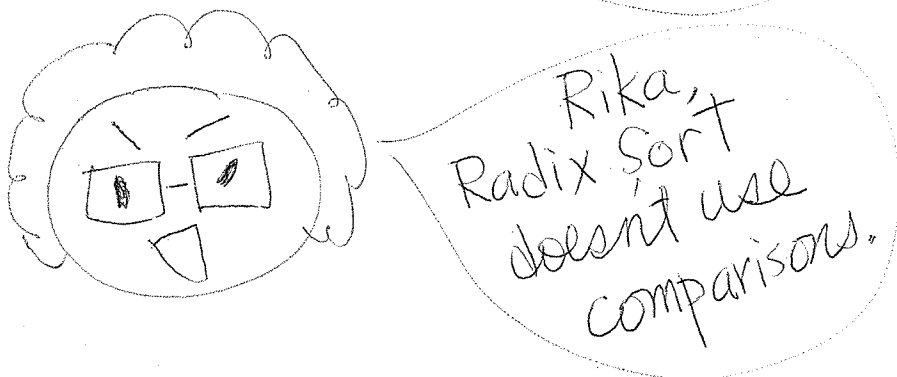
$$B(N) = O(N \log_2 N)$$

All the work
is in Combine



$$F(N) = O(N \log_2 N)$$

comparisons



$$W(N) = O(N \cdot K)$$

