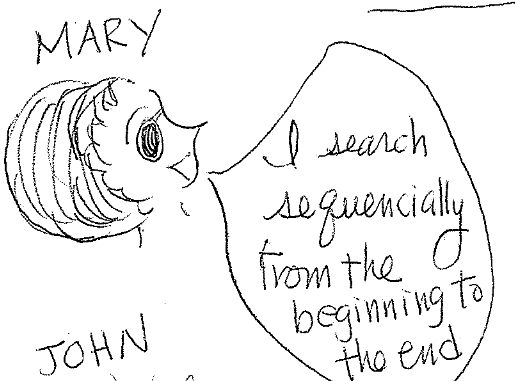
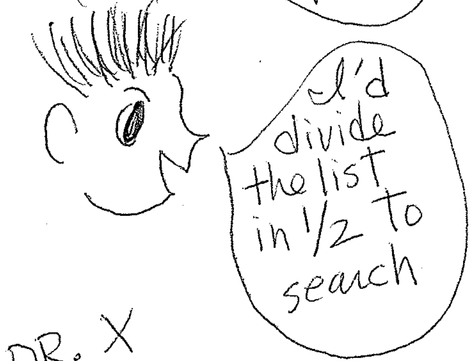
CS311 Analysis Summary



$W(N) = O(N)$
$B(N) = O(1)$
any list



$W(N) = O(\log_2 N)$
$B(N) = O(1)$
sorted list

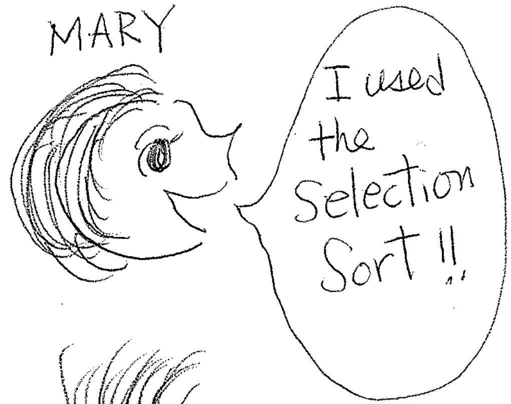
$F(N) = O(\log_2 N)$

search a sorted list

$F(N) = O(N)$

N-1 must lose

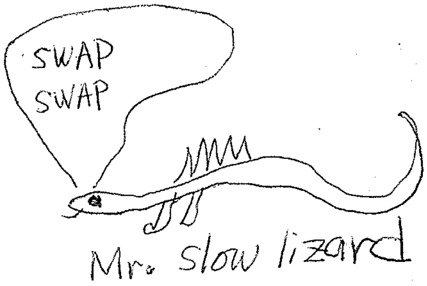
SORTing

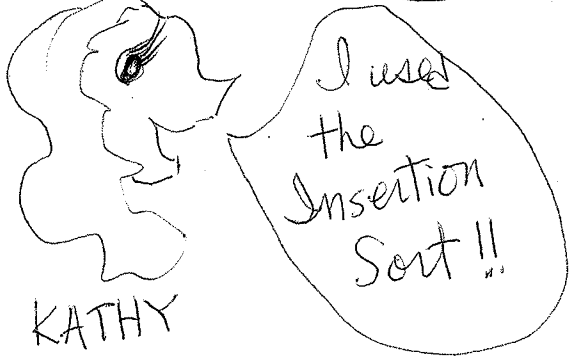


$W(N)=O(N^2)$
$B(N)=O(N^2)$

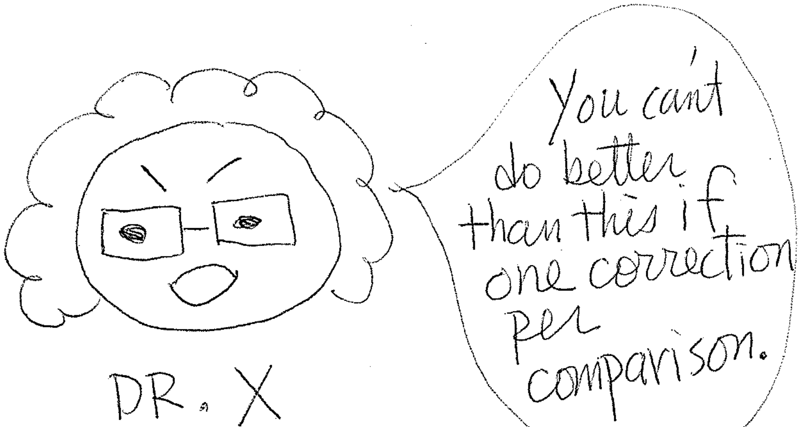
I used the Bubble Sort !!
JOHN

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

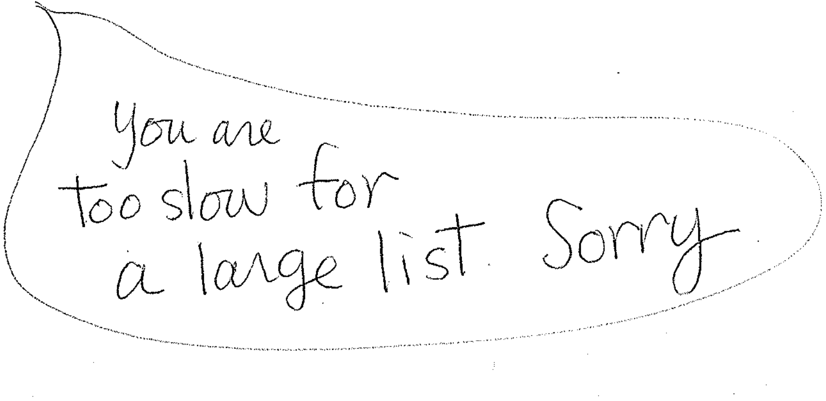




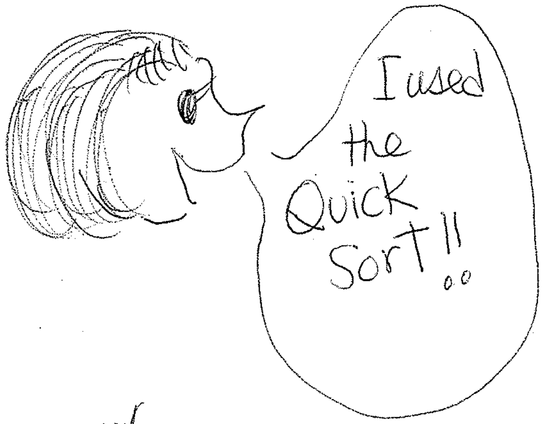
$W(N)=O(N^2)$
$B(N)=O(N)$


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

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

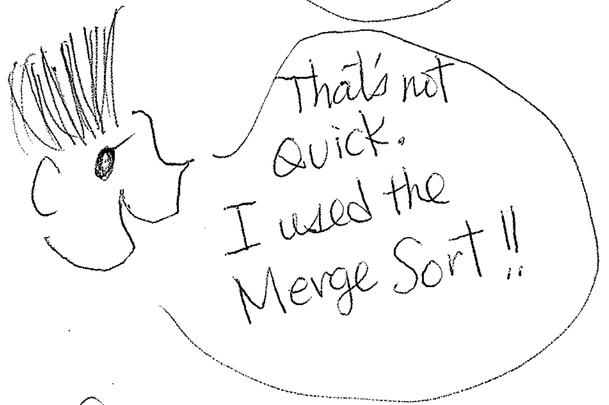


Divide & Conquer Sorting



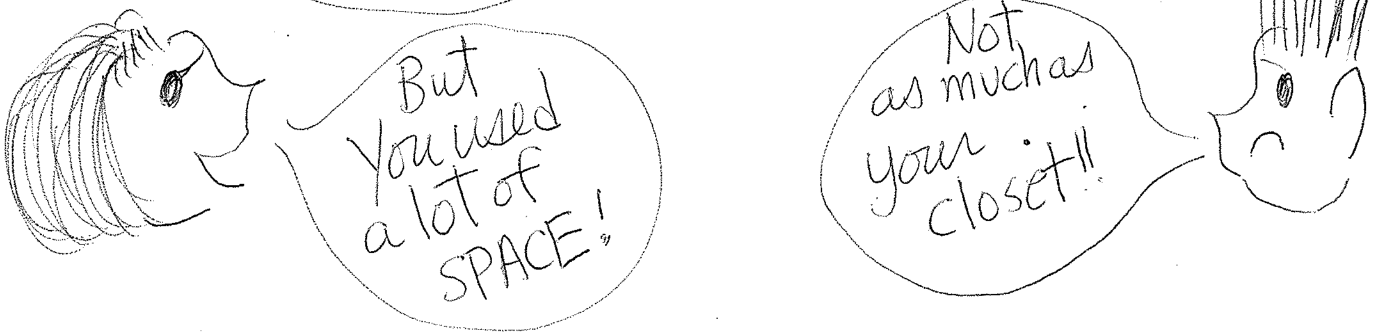
$W(N) = O(N^2)$
$B(N) = O(N \log_2 N)$

Partition ½ each time
vs.
1 & N-1 each time

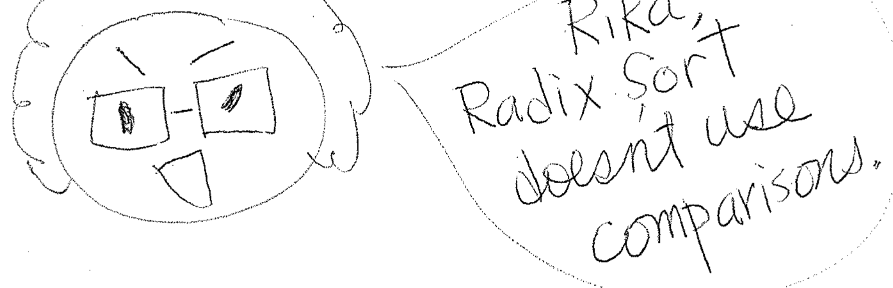


$W(N) = O(N \log_2 N)$
$B(N) = O(N \log_2 N)$

All the work is in Combine



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



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