MergeSort:

T(1) = O(1)

T(n) = 2 T(n/2) + *O(n)*

*=* 2[**2 T(n/4) + n/2**]+n

= 4 T(n/4) + 2n

= 4 [**2 T(n/8) + n/4**] + 2n | n/2^k = 1

= 8 T(n/8) + 3n | n = 2^k

= 2^k T(n/2^k) + kn | k = log2n

= [2^(log2n)]T(1) + nlog2n

= n+nlog2n

= o(nlogn)

n^2 sorts (including selection)

= T(n) = T(n-1) + O(n)

nlogn sorts (avg.case quicksort)

= T(n) = 2 T(n/2) + O(n)

-------------------------------------

T(0) = c

T(n) = T(n-1)+c

= T(n-2)+2c

= T(n-3) + 3c

= T(n-4)+4c

= T(n-k) + kc

= T(n-n) + nc

= c+nc

= O(n)

-------------------------------------

T(0) = c

T(1) = c

T(n) = T(n/2) + c

T(n) = T(n/2) + c

= T(n/4) + 2c |n/2^k = 1

= T(n/8) + 3c |n = 2^k

= t(n/16) + 4c |logn = k

= T(n/2^k) + kc

= T(n/2^lgn) + lgn x c

^^too small

O(lgn)

---------------------------------------

T(0) = c

T(1) = c

T(n) = 2T(n/2) + c

= 4T(n/4) + 3c n/2^k = 1

= 8T(n/8) + 7c n = 2^k

= 16T(n/16) + 15c lgn = k

= 2^kT(n/2^k) + (2^k − 1)c

= 2^lgn T(n/2^lg n) + (2^lg n − 1)c

= nT(n/n) + (n − 1)c

= nT(1) + (n − 1)c

= nc + (n − 1)c

O(n)

T(n) = T(n-1)= n(n+1)/2

T(n) = 2T(n/2) + cn = mergeSort = O(nlgn)

O(n^2) = selection sort

O(2^n) = tower