2-1 Insertion sort on small arrays in merge sort

a. Show that insertion sort can sort the n=k sublists, each of length k, in .nk/ worst-case time.

Answer

Given worst-case of INSERTION-SORT is $f(n) = an^2 + bn + c$, to sort n/k sublists of length k the worst-case is $n/kf(k)$ so:

$$T(n) = \frac{n}{k} f(k) = \frac{n}{k}(ak^2 + bk + c) = ank + bn + c \implies \implies \Theta(nk)$$

b. Show how to merge the sublists in $\Theta(n \lg(n/k))$ worst-case time.

Answer

Considering that $2^i = n/k$ is the number of leaves, so the number of recursions tree is $i = \lg(n/k)$. For each recursion level, it was proved that it takes cn, so the worst-case is

$$cn \lg(n/k) \implies \Theta(n \lg(n/k)) q.e.d.$$

c. Given that the modified algorithm runs in, $\Theta(nk + n \lg(n/k))$ worst-case time, what is the largest value of k as a function of n for which the modified algorithm has the same running time as standard merge sort, in terms of Θ-notation?

Answer

...

d. How should we choose k in practice?

Answer

We should choose k based on statistics input, because the average case of inputs might not be always the worst-case time. So, it is possible that in practice the k has a different value.