11.2-3.

Professor Marley hypothesizes that he can obtain substantial performance gains by modifying the chaining scheme to keep each list in sorted order. How does the professor's modification affect the running time for successful searches, unsuccessful searches, insertions, and deletions?

Answer.

The average-case running time for successful search remains unchanged, which is still $\Theta(1+\alpha)$. The number of elements examinded during a successful search for an element x is one more than the number of elements that smaller than x in x's list. The proof is similar to that of Theorem 11.2 in the text.

Unsuccessful search still takes average-case time $\Theta(1+\alpha)$, but half of the original one. A searching failure arises when k is smaller than the key of the head of $T[h(k)]$, or greater than the key of the tail of $T[h(k)]$, or "sanwich" between keys of two adjcent elements in $T[h(k)]$. This failure is equally likely occurs at any of the position in the list $T[h(k)]$. Thus, the expected number of elements examinded in an unsuccessful search is $E\left[\frac{1}{2}n_{h(k)}\right] = \frac{1}{2}E[n_{h(k)}] = \frac{\alpha}{2}$, and the total time required (including the time for computing $h(k)$) is $\Theta(1+\alpha)$, but reduced a half compared to the unsorted order list in the text.

Insertion in this scenario takes average-time $\Theta(1+\alpha)$, a subtle decline in performance compared to the original of $\Theta(1)$. This is because we need to find the right position instead of the head to insert the new element so that the list remains sorted. Since the major work in insertion is similar to unsuccessful search, they share almost the same average-case time, which is $\Theta(1+\alpha)$.

Deletion is almost the same as successful search, therefore taking average-case time $\Theta(1+\alpha)$.

Email address: informlarry@gmail.com