Math/CoSc 19 supplement, November 12, 2007

This supplement is to clear up some loose ends from class.

First, we consider hashing n keys to m slots. The average population per slot is $\alpha = n/m$. Let n_i be the population in the ith slot, so that

$$\sum_{i=0}^{m-1} n_i = n, \qquad \frac{1}{m}\sum_{i=0}^{m-1} n_i = \alpha.$$

That is, α is the average population per slot.

We ask whether a given object x is one of the keys. We do this by computing $h(x)$, and then looking through the linked list in the slot that $h(x)$ falls into. The time to compute $h(x)$ is 1. If x is not a key, then it is equally likely to land in any slot, so the expected time to look through the linked list is

$$\sum_{i=0}^{m-1} p(x \text{ hashes to slot } i) n_i = \frac{1}{m}\sum_{i=0}^{m-1} n_i = \frac{n}{m} = \alpha.$$

Thus the expected time to decide that x is not a key, when in fact it is not a key, is $\Theta(1+\alpha)$.

Now suppose that x is indeed a key. Now the probability that x hashes to slot i is no longer uniform, it is n_i/n. After computing $h(x)$, the expected time to come to a conclusion about x is

$$\sum_{i=0}^{m-1} p(x \text{ hashes to slot } i) n_i = \sum_{i=0}^{m-1} \frac{n_i}{n} n_i = \frac{1}{n}\sum_{i=0}^{m-1} n_i^2. \tag{1}$$

(So far this is exactly what we did in class.) Let $X_{j,k}$ be the indicator random variable

$$X_{j,k} = \begin{cases} 1, \text{ if key } j \text{ and key } k \text{ map to the same slot,} \\ 0, \text{ if not.} \end{cases}$$

Thus,

$$p(X_{j,k} = 1) = \begin{cases} 1/m, & \text{if } j \neq k, \\ 1, & \text{if not.} \end{cases}$$

Note that the sum over all j and k of $X_{j,k}$ is the number of ordered pairs j, k which map to the same slot. But n_i^2 is the number of ordered pairs of keys which map to slot i, so

$$\sum_{i=0}^{m-1} n_i^2 = \sum_{j,k=0}^{n-1} X_{j,k},$$

where for the sake of simplicity, we assume the keys are numbered 0 to $n-1$. So, we can now find the expectation of $\sum_{i=0}^{m-1} n_i^2$. It is

$$\sum_{j,k=0}^{n-1} p(X_{j,k} = 1).$$

There are $n^2 - n$ pairs j, k with $j \neq k$ and n pairs where $j = k$, so this last sum is

$$(n^2 - n)/m + n < n^2/m + n.$$

Thus, from (1), after computing $h(x)$ the time to search for x is less than

$$\frac{1}{n}\left(\frac{n^2}{m} + n\right) = \alpha + 1.$$

Thus, the total expected time is $< \alpha + 2$, which is $\Theta(1 + \alpha)$.

The other loose end dealt with the analysis of the expected running time of the randomized Quicksort algorithm. Let $\alpha = 5/4$, $\beta = 1/10$, and $\gamma = 9/10$. Then we had shown that

$$T(n) \leq \alpha n + T(\beta n) + T(\gamma n). \tag{2}$$

We'd like to show that there is some $c > 0$ such that $T(n) < cn \log n$ for all large numbers n. For $n = 2$ the expression $n \log n$ is 2, so that we can force the inequality to work for $n = 2$ by taking c large enough, since $T(2) = O(1)$. Let us also assume that c is so large that $\alpha + c(\beta \log \beta + \gamma \log \gamma) \leq 0$. (Note that $\beta \log \beta + \gamma \log \gamma$ is negative.) For example $c \geq 3$ is sufficient here. Assume the inequality $T(m) < cm \log m$ works for numbers m smaller than n. In particular it works for βn and γn, since they are smaller than n. Then by (2),

$$\begin{aligned} T(n) &\leq \alpha n + c\beta n \log(\beta n) + c\gamma n \log(\gamma n) \\ &= n(\alpha + c\beta \log \beta + c\gamma \log \gamma) + (c\beta + c\gamma) n \log n \\ &\leq cn \log n, \end{aligned}$$

where we used $\beta + \gamma = 1$ for the last inequality. We also used the fact that $\log(uv) = \log u + \log v$. Thus, the inequality holds for n as well, so that by induction, it holds for all $n \geq 2$. [This argument could be made a bit more rigorous by worrying about the facts that βn and γn might not be integers, so that $T(\beta n)$ and $T(\gamma n)$ are not defined. This can be remedied by replacing βn with $\lfloor \beta n \rfloor$ and similarly for γn. The same proof works.]