CMSC 641, Design and Analysis of Algorithms, Spring 2010

Sandipan Dey,
Quiz - 2

March 22, 2010

Insertion / Search

Let S denote the sorted array and U denote the unsorted array of maximum size $\sqrt{n}$.

Let there be sequence of n operations, out of which m are INSERT and $n-m$ are SEARCH operations, in any arbitrary order.

Worst case Analysis

Since the cost of any arbitrary operation in the worst case can be at most $n+\sqrt{n}.log\sqrt{n}$ (corresponds to the extreme INSERT case when the sorted array S has size $n-\sqrt{n}$ elements and the unsorted array U has size $\sqrt{n}-1$ elements). Hence the total cost of sequence of n operations $= n(n+\sqrt{n}.log\sqrt{n}) = \theta(n^2)$. Hence, each operation has worst case cost of $\theta(n)$.

Amortized Analysis

Aggregate Method

We notice that INSERT always happens on the unsorted array U first (which is $\theta(1)$) and a subsequent sort on U followed by merge with the sorted array S is required iff U is full after insertion, i.e., if the INSERT operation is $(k\sqrt{n})^{th}$ operation, $k = 1, 2, \ldots \left\lfloor \frac{m}{\sqrt{n}} \right\rfloor$ and there can be only $\left\lfloor \frac{m}{\sqrt{n}} \right\rfloor$ such operations, with actual cost $\sqrt{n}.log\sqrt{n}$ for sorting U and then $k.\sqrt{n}$ for merging with S, a total of $\theta\left(\sqrt{n}.log\sqrt{n} + k.\sqrt{n}\right)$ for the $(k\sqrt{n})^{th}$ operation.

Also, SEARCH always happens first on the sorted array S (binary search, can at most take $\theta(lgn)$ time) and if not found then it proceeds to unsorted array U (linear search, (can at most take $\theta(\sqrt{n})$ time)), hence the total time per SEARCH is at most $\theta(logn) + \theta(\sqrt{n}) = \theta(\sqrt{n})$.

Hence the actual cost of all m INSERT operations

$$= \left(m - \left\lfloor \frac{m}{\sqrt{n}} \right\rfloor\right) .\theta(1) + \sum_{k=1}^{\left\lfloor \frac{m}{\sqrt{n}} \right\rfloor} \theta\left(\sqrt{n}.log\sqrt{n} + k.\sqrt{n}\right)$$

$$= \theta\left(m - \left\lfloor \frac{m}{\sqrt{n}} \right\rfloor\right) + \theta(m.log\sqrt{n}) + \theta\left(\sqrt{n}. \sum_{k=1}^{\left\lfloor \frac{m}{\sqrt{n}} \right\rfloor} k\right)$$

$$=\theta\left(m - \left\lfloor \frac{m}{\sqrt{n}} \right\rfloor + m.log\sqrt{n}\right) + \theta\left(\sqrt{n}.\frac{\left\lfloor \frac{m}{\sqrt{n}} \right\rfloor.\left(\left\lfloor \frac{m}{\sqrt{n}} \right\rfloor + 1\right)}{2}\right)$$

$$= \theta\left(\frac{m^2}{\sqrt{n}}\right) = \theta\left(m\frac{n}{\sqrt{n}}\right) = \theta(m\sqrt{n}),$$ since m can be at most $\theta(n)$.

Hence amortized cost per INSERT operation $= \theta(\sqrt{n})$.

Similarly, actual cost of $n-m$ SEARCH operations will eb at most $(n-m)\theta(\sqrt{n})$, amortized cost per operation being $\theta(\sqrt{n})$.

Accounting Method

We notice that there can be at most $\left\lfloor \frac{m}{\sqrt{n}} \right\rfloor$ INSERT operations that can take at most $\theta\left(\sqrt{n}.log\sqrt{n} + n\right) = \theta\left(\sqrt{n}.\left(\sqrt{n} + log\sqrt{n}\right)\right)$ time.

Let's charge every INSERT operation by $(\sqrt{n} + log\sqrt{n} + 1)$ \$, of which 1\$ is acutually required per INSERT and the rest stored on the element in U for the future operations.

After $\sqrt{n}$ such INSERT operations when U is full, we already have $\sqrt{n}.(\sqrt{n} + log\sqrt{n}) = (n + \sqrt{n}.log\sqrt{n})$ \$ charge stored on the elements of U, $\sqrt{n}.log\sqrt{n}$ \$ will be spent in sorting U and $\sqrt{n}$ \$ is to be spent in merging U with S, the rest (i.e., $(n - \sqrt{n})$ \$) is to be stored on the elements on S, precisely $(\sqrt{n} - 1)$ \$ on each element of S. But in future there can be at most $\sqrt{n} - 1$ merge operations, hence each element can provide its cost for merging.

Similarly charge $\sqrt{n}$ \$ for each SEARCH operation.

Potential Method

Intuitively we can see that the potential should increase as the size of U increases, in order to accomodate the cost for future sorting/merging. Let's define the potential as $\phi = |U|\left(lg|U| + \frac{|S|}{\sqrt{n}}\right)$, where $|U|$ and $|S|$ denote the numberof elements in U and S respectively.

Let's start with $|S| = s = (k-1)\sqrt{n},\ k = 1, 2,\ \sqrt{n}$, immediately before the i^{th} operation.

If initially $|U| = u < \sqrt{n} - 1$ and the i^{th} operation is INSERT, we have
$\phi_{i-1} = u\left(lg\ u + \frac{s}{\sqrt{n}}\right)$.
$\phi_i = (u+1)\left(lg(u+1) + \frac{s}{\sqrt{n}}\right)$, s.t.
$\Rightarrow \phi_i - \phi_{i-1} = (u+1)lg(u+1) - ulgu + \frac{s}{\sqrt{n}} = u.lg\ (1 + \frac{1}{u}) + lg(u+1) + \frac{(k-1)\sqrt{n}}{\sqrt{n}}$
$= u.\theta(1) + \theta(lg\ u) + \theta(k) = \theta(u + lg\ u + k) = \theta(\sqrt{n})$, since $1 \leq k \leq \sqrt{n}$ and $0 \leq u < \sqrt{n} - 1$.
But actual cost of INSERT here $= \theta(1)$. Hence, $\hat{c_i} = c_i + \phi_i - \phi_{i-1} = \theta(\sqrt{n})$.

Similarly, if initially $|U| = u = \sqrt{n} - 1$, we shall have $u = 0$ and $s = k\sqrt{n}$ after INSERT. Since we have
$\phi_{i-1} = u\left(lg\ u + \frac{(k-1)\sqrt{n}}{\sqrt{n}}\right) = (\sqrt{n} - 1)\left(lg\ (\sqrt{n} - 1 + k - 1)\right) = \theta(\sqrt{n}lg\ \sqrt{n}) + (k-1)\sqrt{n}$ and
$\phi_i = 0$
$\Rightarrow \phi_i - \phi_{i-1} = -\theta(\sqrt{n}lg\ \sqrt{n}) - (k-1)\sqrt{n}$
But actual cost of INSERT here $= \theta(\sqrt{n}lg\ \sqrt{n}) + k\sqrt{n}$. Hence, $\hat{c_i} = c_i + \phi_i - \phi_{i-1} = \theta(\sqrt{n})$.