CLRS 17-2

Peter Danenberg

October 15, 2008

a. Search through the preliminary elements of $A_0, \ldots, A_{\lg n - 1}$, finding $A_i[0] \leq x$; search A_i for x: if x is not found, search the next $A_{i<j<\lg n}$ such that $A_j[0] \leq x$ or declare "not found."

   Worst case, must search through all $A_0, \ldots, A_{\lg n}$ arrays in $\Theta(\lg n)$; each of which must be binary-searched in $\Theta(\lg n)$ (since already sorted): for a total of $\Theta(\lg^2 n)$.

b. Insert an element as follows:

```
INSERT(A, x)
if ⌈lg n + 1⌉ > k
    then Allocate new array A_{k+1}
         k ← k + 1
while i < k and n[i] = 1
      do MERGE(A_i, A_{i+1})
         n[i] ← 0
         i ← i + 1
n[i] ← 1
```

   Worst case merge and remerge arrays in $\Theta(2^{\lg n}) \in \Theta(n)$ time; alternatively, merge sorted lists in $\Theta(n)$:

$$\sum_{i=0}^{k-1} 2^i = \left(\frac{2^{k+1} - 1}{2 - 1}\right) \tag{1}$$

$$\in \Theta(2^k) \tag{2}$$

$$\in \Theta(n) \tag{3}$$

   Since, for every bit set, there are worst-case $\Theta(\lg n)$ merge operations; we can define modify the potential function in the counter example followingly:

$$\Phi(D_i) - \Phi(D_{i-1}) \leq \lg n \cdot (b_{i-1} - t_i + 1) - \lg n \cdot b_{i-1} \tag{4}$$

$$= \lg n \cdot (t_i - 1) \tag{5}$$

And analyze the amortized complexity:

$$\begin{aligned}
\hat{c}_i &= c_i + \Phi(D_i) - \Phi(D_{i-1}) && (6)\\
&= \lg n \cdot t_i + \lg n \cdot (1 - t_i) && (7)\\
&= \lg n && (8)
\end{aligned}$$

The amortized cost of each INSERT is therefore $\Theta(\lg n)$.

c. For DELETE(x): find A_j, the first full array; and A_i, the array to which x belongs. Remove an element y from A_j and replace x in A_i with it; distribute the remaining items from A_j into $A_0, \ldots, A_{j-1}$.

   Worst case analysis: $\Theta(\lg n)$ to find A_j; $\Theta(\lg^2 n)$ to find A_i; $\Theta(2^{\lg n}) \in \Theta(n)$ to insert y into A_i; and $\Theta(2^{\lg n-1}) \in \Theta(n)$ to distribute A_j to its underarrays: for $\Theta(n)$ total in the worst case.