CSC 263 Friday Tutorial Mar 9, 2018

→ Topic: Amortized Analysic, Sample Assignment Q

- Data structure representing a set I of integers; doubly linked list of arrays such that
  - (i) each element of I occurs exactly once in the list
  - (ii) each array is sorted
  - (iii) the number of elements in each array is a power of 2
  - (iv) no 2 arrays in the list have the same size
  - (v) arrays in linked list kept in order of increasing size
  - (vi) first element of each array stores SIZE field.

Ex: $I = \{3, 5, 1, 17, 10\}$: head ⟷ [3] ⟷ [1, 5, 10, 17] ⟷ tail

Ex: $I = \{17, 8, 3, 10, 1, 12, 6\}$:

head ⟷ [17] ⟷ [3, 8] ⟷ [1, 6, 10, 12] ⟷ tail

- SEARCH(X) // high-level pseudocode
  - binary search on arrays until you find x or until search is exhaused

- Worst case performance of SEARCH:
  - x is not in I
  - max # of arrays, $n = 2^i - 1$ for some $i \in \mathbb{N}$, $\log n$ arrays
  - must perform binary search on heaps of size 2^i, $0 \le i \le \log n$
  - worst case binary search on heap of size $2^i = \log_2(2^i)$
  - $\therefore$ Worst case $= \sum_{i=0}^{i=\log n} \log_2 2^i = \sum_0^{\log n} i \in \Theta(\log^2 n)$

- INSERT(X)
  - create new array of size 1
  - while linked list contains multiple trees of same size, merge

[sub procedure of merge sort]

[TODO: learn about merge]

- Worst-case performance of INSERT:
  - occurs when most merges take place
  - most merges when heap contains arrays of all sizes
  - worst case when $n = 2^i - 1$ where $i \in \mathbb{N}$
  - merge is $\Theta(n)$, merges on all $\log n$ arrays of size 2^i
  - $\therefore$ runtime $= \sum_{i=0}^{\log n} 2^i = 2^{\log n + 2} - 1 = 2^2 2^{\log n} - 1 = 4n - 1 \in \Theta(n)$
  - // Note: closed form of $\sum_{i=0}^{k} 2^i$ is ??? used in proof above.

- Why amortized analysis?
  - often data structures are expected to take a sequence of operations: when worst-cases are known to be rare worst-case analysis doesnt tell us much about real results

- Amortized analysis of INSERT
  - "suppose we execute a sequence of n INSERTS starting from empty set I. Determine the upper bound on amortized time."
  - "insertion costs at most $O(2^r)$ where r is the position of the first 0 digit in the binary representation of n"
    - [Have to merge all trees of size 2^i for $0 \leq i < r$ (where there is a 1 in bin rep of n), $\sum_{i=0}^{r-1} 2i < 2^r \therefore O(2^r)$
  - $r = 0$ happens $\lceil n/2 \rceil$ times, $(r=1) \to (\lceil n/4 \rceil)$, $(r=2) \to (\lceil n/8 \rceil)$
  - $\sum_{r=0}^{\lceil \log n \rceil} \lceil n/2^{r+1} \rceil \times 2^r = n \sum_{r=0}^{\lceil \log n \rceil} 1 = n \times \lceil \log n \rceil \in O(n \log n)$