Tutorial 6, Oct 27, 2023

- Consider an ADT storing a multiset with the operations INSERT(x), which inserts an element, and DIMINISH(), which deletes the $\left\lceil \frac{n}{2} \right\rceil$ largest elements
- We will use an unsorted linked list and n which tracks the size
  - On each insert we simply add x to the end of the list and increment n
  - On diminish, we find the median ($5n$ comparisons), and then loop over all elements deleting elements greater than m and enough copies of m
- We will show that this is $O(1)$ amortized in the number of comparisons
  - Each insert costs no comparisons, but we charge 12
  - Each diminish costs $6n$ comparisons ($5n$ for median finding and another n to delete), but we charge 0
- Invariant: for all sequences of operations of length l: credit stored in each element is 12, and the total charge is at least the total actual cost
  - Prove by induction
  - Base case is for a length 0 operation, in which case this is trivially true
  - Inductive step: assuming this holds for $l - 1$, we will prove for l
    * Case 1: the new operation is an insert
      - We charge 12, so the new element has credit 12
      - We charge 12 but use nothing, so the second part also holds
    * Case 2: the new operation is a diminish
      - We will pay for this operation with the credit of the deleted elements
      - Each deleted element has 12 credits, so we have $\left\lceil 12\frac{n}{2} \right\rceil = 6n$ credits to work with
      - But each diminish takes exactly $6n$ as explained above, so the credit invariant is maintained
      - We also don't touch the credits on the undeleted elements so the first invariant also holds
- Therefore a sequence of length l uses at most $12l$ charge, so each operation is $O(1)$