Assignment1 Q1 Wrote Muzi Zhao /Read Jingbo Yang

(a) Define Costs and Charges

We know the cost of an operation is defined by the number of pairwise comparisons between the elements of S that the operation takes to execute. The cost of Add is 0, and the cost of Reduce(S) is αn.

- For Add(S, x): charge new element x with 2α. When adding a new element x to the bag S, we assign a charge of 2α credits to the Add operation.
- For Reduce(S): charge 0. Instead of assigning new credits at the time of reduction, we rely on the credits that were accumulated during previous Add operations to cover the cost of this operation.

(b) Credit Invariant

Credit Invariant CI(k): each element in S has a credit of at least 2α after the k-th step in a sequence of m operations.

Base case: When $k = 0$, CI(0) is vacuously true.

Inductive step: Assume CI(k) holds. We want to show that CI($k+1$) holds for $k > 0$. We consider different cases:

(i) If at the $k+1$-th step we do an Add, according to our assumption, any node in our linked list has $c \geq 2\alpha$. Therefore, if we add one more node with $c = 2\alpha$, CI($k+1$) holds.

(ii) If at the $k+1$-th step we do a Reduce, we will decrease the credit stored in each node by α. Therefore, the total credit paid would be αn, which includes $(\alpha - 1)n$ for *MEDIAN* and n for the reduce operation. For each node, $c \geq 2\alpha - \alpha = \alpha$; thus, total credit $\geq \alpha n$. After actually deleting the largest integers, the remaining length would be $n - \lfloor \frac{n}{2} \rfloor \leq \frac{n}{2}$. Therefore, at this time, for each node $c \geq \frac{\alpha n}{\frac{n}{2}} = 2\alpha$.

That is to say, CI($k+1$) holds.

(c) Amortized Cost Calculation

In a sequence of ADD and REDUCE operations, we use the accounting method to calculate the amortized cost.

- After N consecutive ADD operations, the size of S becomes N.
- We can have at most $\log(N)$ REDUCE operations, since each REDUCE reduces the size of S by at least half. Thus, in the worst case, the total number of operations is $N + \log(N)$. By the credit invariant (CI), we know that each element always has a credit of 2α. Each Reduce operation costs αN. After N Add operations, the total credit is $2\alpha N$, which covers the cost of all Reduce operations, since $2\alpha N - \alpha N \log N > 0$.

The amortized cost per operation is:

$$\text{Amortized Cost} = \frac{2\alpha N}{N + \log(N)} = O(2\alpha)$$

This shows that the amortized cost is $O(1)$, since α is a constant.