Lab 3

Solution

1. Suppose we perform a sequence of n operations on a data structure in which the i-th operations cost i if i is an exact power of 2 and 1 otherwise. Use both Aggregate analysys method and Amortized analysis method to determine the amortized cost per operation.

Ans.

(a) Aggregate matheod:

Let c_i = cost of ith operation.

$$c_i = \begin{cases} i & \text{if } i \text{ is an exact power of } 2 , \\ 1 & \text{otherwise} . \end{cases}$$

Operation	Cost
1	1
2	2
3	1
4	4
5	1
6	1
7	1
8	8
9	1
10	1
⋮	⋮

n operations cost

$$\sum_{i=1}^{n} c_i \le n + \sum_{j=0}^{\lg n} 2^j = n + (2n-1) < 3n .$$

(Note: Ignoring floor in upper bound of $\sum 2^j$.)

Average cost of operation $= \dfrac{\text{Total cost}}{\text{\# operations}} < 3$

By aggregate analysis, the amortized cost per operation $= O(1)$.

(b) Accounting Method

Let c_i = cost of ith operation.

$$c_i = \begin{cases} i & \text{if } i \text{ is an exact power of } 2 , \\ 1 & \text{otherwise} . \end{cases}$$

Charge each operation \$3 (amortized cost $\widehat{c_i}$).

- If i is not an exact power of 2, pay \$1, and store \$2 as credit.
- If i is an exact power of 2, pay \$$i$, using stored credit.

Operation	Cost	Actual cost	Credit remaining
1	3	1	2
2	3	2	3
3	3	1	5
4	3	4	4
5	3	1	6
6	3	1	8
7	3	1	10
8	3	8	5
9	3	1	7
10	3	1	9
⋮	⋮	⋮	⋮

Since the amortized cost is \$3 per operation, $\sum_{i=1}^{n} \widehat{c_i} = 3n$.

We know from Exercise 17.1-3 that $\sum_{i=1}^{n} c_i < 3n$.

Then we have $\sum_{i=1}^{n} \widehat{c_i} \geq \sum_{i=1}^{n} c_i \Rightarrow$ credit = amortized cost − actual cost ≥ 0.

Since the amortized cost of each operation is $O(1)$, and the amount of credit never goes negative, the total cost of n operations is $O(n)$.

Problems #2 to #4

See Java codes.