Week – 3 Graded Homework (Solutions)

Solution 1:

Aggregate Method:

Total cost $T = \sum_{i=0}^{\lfloor \log n \rfloor}(2^i) + (n - \lfloor \log n \rfloor - 1)$

$$\begin{aligned} T &= \sum_{i=0}^{\lfloor \log n \rfloor}(2^i) + (n - \lfloor \log n \rfloor - 1) \\ &= 3n - \lfloor \log n \rfloor - 2 \\ &= O(n) \end{aligned}$$

Hence the amortized cost is $O(1)$.

Accounting Method:

Let the amortized cost be

$$\begin{aligned} &1, \text{ if } i = 1 \text{ i.e., } i = 2^0 \\ &2, \text{ if } i = 2^k \text{ for all } k \geq 1 \\ &3, \text{ otherwise} \end{aligned}$$

The actual cost is

$$\begin{aligned} &1, \text{ if } i = 1 \text{ i.e., } i = 2^0 \\ &i, \text{ if } i = 2^k \text{ for all } k \geq 1 \\ &1, \text{ otherwise} \end{aligned}$$

After first and the second operation, there is no credit left since their actual costs are the same as their amortized costs. For every jth operation where j is not an exact power of 2 (which only cost 1), two extra credit is saved for later use. For every operation i where $i = 2^k$. Its actual cost is $i = 2^k$. We can use the credits saved before the ith operation. Since there are $2(2^k - 2^{k-1} - 1)$ credits saved, and since we charge two for this operation, we can pay for the actual cost of the ith operation. Thus the total amortized cost is an upper bound on the total actual cost of any sequence of operations. Hence the amortized cost is $O(1)$.

Potential Method:

Define the following potential function:

$$\begin{aligned} &\Phi(D_0) = 0 \\ &\Phi(D_i) = 2i - 2^p \text{ where } 2^{p-1} \leq i < 2^p \end{aligned}$$

Then $\Phi(D_i) \geq 0 = \Phi(D_0) = 0$. If i is exact power of 2, let $i = 2^k$

$$\begin{aligned} \hat{c}_i &= c_i + \Phi(D_i) - \Phi(D_{i-1}) \\ &= i + \{2i - 2^{k+1} - 2(i-1) - 2^k\} \\ &= 2 = O(1) \end{aligned}$$

If i is not exact power of 2,

$$\begin{aligned} \hat{c}_i &= c_i + \Phi(D_i) - \Phi(D_{i-1}) \\ &\leq 1 + \{2i - 2^p - 2(i-1) - 2^p\} = O(1) \end{aligned}$$

Hence the amortized cost is $O(1)$.

Solution 2:

Answer: We define the potential function Φ on a stack to be the number of objects in the stack. For the empty stack D_0 with which we start, we have $\Phi(D_0) = 0$. Since the number of objects in the stack is never negative, the stack D_i that results adter the i^{th} operation has non-negative potential, and thus $\Phi(D_i) \geq 0 = \Phi(D_0)$.

The total amortized cost of n operations with respect to Φ therefore represents an upper bound on the actual cost.

Let us now compute the amortized cost of the various stack operations. If the i^{th} operation on a stack containing s objects is a PUSH operation, then the potential difference is $\Phi(D_i) - \Phi(D_{i-1}) = (s+1) - s = 1$. Hence the amortized cost of the PUSH operation is $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = 1 + 1 = 2$.

Suppose that the i^{th} operation on the stack is MULTIPOP(S, k), which cause $k' = min(k, s)$ objects to be popped off the stack. The actual cost of the operation is k' and the potential difference is $\Phi(D_i) - \Phi(D_{i-1}) = -k'$. Thus, the amortized cost of the MULTIPOP operation is $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = k' - k' = 0$.

Similarly, the amortized cost ofan ordinary POP operation is 0.

The amortized cost of each of the three operations are $O(1)$, and thus the total amortized cost of a sequence of n operations is $O(n)$. Since we have already argued that $\Phi(D_i) \geq \Phi(D_0)$, the total amortized cost of n operations is an upper bound on the total actual cost. The worst-case of n operations is there fore $O(n)$.

Solution 3:

Let the potential function $\Phi = \sum_{v \in Heap} depth(v)$, where $depth(v)$ means that the number of edges of the (shortest) path from the root to node v.

Then $\Phi_0 = 0$, since there is no node in the heap. Since the number of edges in the heap is never negative, $\Phi_i \geq \Phi_0$.

Suppose k-th operation is INSERT. Then

$$\begin{aligned}
\hat{c}_k &= c_k + \Phi_k - \Phi_{k-1} \\
\hat{c}_k &\leq \lceil \log n \rceil + \lceil \log n \rceil \\
&\leq 2\lceil \log n \rceil = O(\log n)
\end{aligned}$$

Suppose k-th operation is EXTRACT-MIN. Then

$$\begin{aligned}
\hat{c}_k &= c_k + \Phi_k - \Phi_{k-1} \\
\hat{c}_k &\leq (\lceil \log n \rceil + 1) - \lceil \log n \rceil \\
&= 1 = O(1)
\end{aligned}$$