Amortized Analysis Part-II (DAA, M.Tech + Ph.D.)

By:

Sunil Kumar Singh, PhD
Assistant Professor,
Department of Computer Science and Information Technology

School of Computational Sciences, Information and Communication Technology,
Mahatma Gandhi Central University, Motihari
Bihar, India-845401

Outline

- Amortized Analysis
- An Example
- Types of Amortized Analysis
- Aggregate Analysis
- The Accounting Method
- The Potential Method
- Conclusion
- References

Potential Method

- Instead of representing prepaid work as credit stored with specific objects in the data structure, the **potential method** of amortized analysis represents the prepaid work as "potential energy" or just potential that can be represented to pay for future operations.

- The potential is associated with the data structure as a whole rather than with specific objects within the data structure.

- Working of this method is as follows; we start with an data structure D_0 on which n operations are performed. For each i=1,2,…,n, we let c_i be the actual cost of i^{th} operation and D_i be the data structure that results after applying the i^{th} operation to data structure D_{i-1}.

Potential Method cont..

- A **potential function** Ø maps each data structure D_i to a real number $Ø(D_i)$, which is the **potential** associated with data structure D_i. The amortized cost $\hat{c_i}$ of the i^{th} operation with respect to potential function Ø is defined by

$$\hat{c_i} = c_i + Ø(D_i) - Ø(D_{i-1})$$

- The amortized cost of each operation is therefore its actual cost cost plus the increase in potential due to the operation. The total amortized cost on the n operations is

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} (c_i + \Phi(D_i) - \Phi(D_{i-1}))$$

$$= \sum_{i=1}^{n} c_i + \Phi(D_n) - \Phi(D_0)$$

Potential Method cont..

- If we can define a potential a potential function Ø so that $(D_n)\geq (D_0)$, then the total amortized cost $\sum_{i=1}^{n} \hat{c}_i$ is an upper bound on the total actual cost $\sum_{i=1}^{n} c_i$
- In practice, we do not always know how many operations might be performed. Therefore, if we required that $Ø(D_i)\geq Ø(D_0)$ for all i, then we guarantee, as in the accounting method, that we pay in advance. It is often convenient to define $Ø(D_0)$ to be 0 and then show that $Ø(D_i)\geq 0$ for all i.
- Intuitively, if the potential difference $Ø(D_i) - Ø(D_{i-1})$ of the i^{th} operation is positive, then the amortized cost c_i represents an overcharge to the i^{th} operation, and the potential of the data structure increases. If the potential difference is negative, then the amortized cost represents an undercharge to the i^{th} operation, and the actual cost of the operation is paid by the decrease in the potential.

Potential Method cont..

Example: Incrementing a binary counter

- As an example of the potential method, we again look at incrementing a binary counter. This time, we define the potential of the counter after the i^{th} INCREMENT operation to be b_i , the number of 1's in the counter after the i^{th} operation.

- Let us computer the amortized cost of an INCREMENT operation. Suppose that the i^{th} INCREMENT operation resets t_i bits, it sets at most one bit to 1.

- If $b_i = 0$, then the i^{th} operation resets all k bits, and so $b_{i-1} = t_i = k$.

- If $b_i > 0$, then $b_i = b_{i-1} - t_i + 1$

Potential Method cont..

- In either case, $b_i \le b_{i-1} - t_i + 1$, and the potential difference is
  $\Phi(D_i) - \Phi(D_{i-1}) \le (b_{i-1} - t_i + 1) - b_{i-1} = 1 - t_i$

- The amortized cost is therefore

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

- If the counter starts at zero, then $\emptyset(D_0)$=0. since $\emptyset(D_i) \geq 0$ for all i, the total amortized cost of a sequence of n INCREMENT operations is an upper bound on the total actual cost, and so the worst-case cost of n INCREMENT operations is $O(n)$.

Potential Method cont..

- The potential method gives us an easy way to analyze the counter even when it does not start at zero. There are initially b_0 1's, and after n INCREMENT operations there are b_n 1's, where $0 \leq b_i, b_n \leq k$.(recall that k is the number of bits in the counter.) we can rewrite equation as

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} \widehat{c_i} - \Phi(D_n) + \Phi(D_0)$$

- We have $\hat{c}_i \leq 2$ for all $1 \leq i \leq n$. Since $\Phi(D_0)$=b_0 and $\Phi(D_n)$ =b_n , the total actual cost on n INCREMENT operations is

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} 2 - b_n + b_0$$
$$= 2n - b_n + b_0$$
$$= O(n)$$
$$(\because 0 \leq b_n, b_0 \leq k)$$

Potential Method cont..

- Note in particular that since $b_0 \leq k$, as long as $k = O(n)$, the total actual cost is $O(n)$. In other words, if we execute at least $n = (k)$ INCREMENT operations, the total actual cost is $O(n)$, no matter what initial vale the counter contains.

Exercises

1. Show the analysis of Stack Operation by potential method
2. Analysis of Dynamic tables by the following methods:
    a. Accounting method
    b. Potential method

Conclusions

- Amortized costs can provide a clean abstraction of data-structure performance.
- Any of the analysis methods can be used when an amortized analysis is called for, but each method has some situations where it is arguably the simplest.
- Different schemes may work for assigning amortized costs in the accounting method, or potentials in the potential method, sometimes yielding radically different bounds.

References

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.
2. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. "Introduction to algorithms second edition." *The Knuth-Morris-Pratt Algorithm, year* (2001).
3. Seaver, Nick. "Knowing algorithms." (2014): 1441587647177.

Thank You